This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2019.2930039, IEEE Transactions on Automatic Control

Optimal Secure Control with Linear Temporal Logic Constraints

Luyao Niu, *Student Member, IEEE,* and Andrew Clark, *Member, IEEE*

***Abstract*—Prior work on automatic control synthesis for cyber-physical systems under logical constraints has primarily focused on environmental disturbances or modeling uncertainties, however, the impact of deliberate and malicious attacks has been less studied. In this paper, we consider a discrete-time dynamical system with a linear temporal logic (LTL) constraint in the presence of an adversary, which is modeled as a stochastic game. We assume that the adversary observes the control policy before choosing an attack strategy. We investigate two problems. In the first problem, we synthesize a robust control policy for the stochastic game that maximizes the probability of satisfying the LTL constraint. A value iteration based algorithm is proposed to compute the optimal control policy. In the second problem, we focus on a subclass of LTL constraints, which consist of an arbitrary LTL formula and an invariant constraint. We then investigate the problem of computing a control policy that minimizes the expected number of invariant constraint violations while maximizing the probability of satisfying the arbitrary LTL constraint. We characterize the optimality condition for the desired control policy. A policy iteration based algorithm is proposed to compute the control policy. We illustrate the proposed approaches using two numerical case studies.**

***Index Terms*—Linear Temporal Logic (LTL), stochastic game, adversary.**

I. Introduction

Cyber-physical systems (CPS) are expected to perform increasingly complex tasks in applications including autonomous vehicles, teleoperated surgery, and advanced manufacturing. An emerging approach to designing such systems is to specify a desired behavior using formal methods, and then automatically synthesize a controller satisfying the given requirements [1]–[5].

Temporal logics such as linear temporal logic (LTL) and computation tree logic (CTL) are powerful tools to specify and verify system properties [6]. In particular, LTL, whose syntax and semantics have been well developed, is widely used to express system properties. Typical examples include liveness (e.g., "always eventually A"), safety (e.g., "always not A"), and priority (e.g., "first A, then B"), as well as more complex tasks and behaviors [6]. For systems operating in stochastic environments or imposed probabilistic requirements (e.g., "reach A with probability 0.9"), probabilistic extensions have also been proposed such as safety and reachability games that capture worst-case system behaviors [7].

In addition to modeling uncertainties and stochastic errors [7]–[9], CPS will also be subject to malicious attacks, including denial-of-service and injection of false sensor measurements and control inputs [10], [11]. For instance, power outages have been reported due to the penetration of attackers in power systems [10]. Attacks against cars and UAVs are also reported in [11], [12]. Unlike stochastic errors/modeling uncertainties, intelligent adversaries are able to adapt their strategies to maximize impact against a given controller, and thus exhibit strategic behaviors. Moreover, controllers will have limited information regarding the objective and strategy of the adversary, making techniques such as randomized control strategies potentially effective in mitigating attacks. In this case, control strategies synthesized using existing approaches may be suboptimal in the presence of intelligent adversaries because they are designed for CPS under errors and uncertainties. However, automatic synthesis of control systems in adversarial scenarios has received limited research attention.

In this paper, we investigate two problems for a probabilistic autonomous system in the presence of an adversary who tampers with control inputs based on the current system state. We abstract the system as a stochastic game (SG), which is a generalization of Markov decision process (MDP). We assume a concurrent Stackelberg information structure, in which the adversary and controller take actions simultaneously. Stackelberg games are popular models in security domain [13]–[16]. Turn-based Stackelberg games, in which a unique player takes action each time step, have been used to construct model checkers [17] and compute control strategy [18], however, to the best of our knowledge, control synthesis in the concurrent Stackelberg setting has been less investigated.

We focus on two problems. In the first problem, we are given an arbitrary LTL specification and focus on generating a control strategy such that the probability of satisfying the specification is maximized. In the second problem, we focus on a subclass of LTL specification that combine an arbitrary LTL specification with an invariant constraint using logical and connectives, where an invariant constraint requires the system to always satisfy some property. The specification of interest is commonly required for CPS, where the arbitrary LTL specification can be used to model properties such as liveness and the invariant property can be used to model safety property. We consider the scenario where the specification cannot be satisfied. Hence, we relax the specification by allowing violations on the invariant constraint and we select a control policy that minimizes the average rate at which invariant property violations occur while maximizing the probability of satisfying the LTL specification. We make the following specific contributions:

- We formulate an SG to model the interaction between

L. Niu and A. Clark are with the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA 01609 USA. {lniu,aclark}@wpi.edu

This work was supported by NSF grant CNS-1656981.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2019.2930039, IEEE Transactions on Automatic Control

the CPS and adversary. The SG describes the system dynamics and the effects of the joint input determined by the controller and adversary. We propose a heuristic algorithm to compute the SG given the system dynamics.

- We investigate how to generate a control policy that maximizes the worst-case probability of satisfying an arbitrary specification modeled using LTL. We prove that this problem is equivalent to a zero-sum stochastic Stackelberg game, in which the controller chooses a policy to maximize the probability of reaching a desired set of states and the adversary chooses a policy to minimize that probability. We give an algorithm to compute the set of states that the system desires to reach. We then propose an iterative algorithm for constructing an optimal stationary policy. We prove that our approach converges to a Stackelberg equilibrium and characterize the convergence rate of the algorithm.
- We formulate the problem of computing a stationary control policy that minimizes the rate at which invariant constraint violations occur under the constraint that an LTL specification must be satisfied with maximum probability. We prove that this problem is equivalent to a zero-sum Stackelberg game in which the controller selects a control policy that minimizes the average violation cost and the adversary selects a policy that maximizes such cost. We solve the problem by building up the connections with a generalized average cost per stage problem. We propose a novel algorithm to generate an optimal stationary control policy. We prove the optimality and convergence of the proposed algorithm.
- We evaluate the proposed approach using two numerical case studies in real world applications. We consider a remotely controlled UAV under deception attack given different LTL specifications. We compare the performance of our proposed approaches with the performance obtained using existing approaches without considering the adversary's presence. The results show that our proposed approach outperforms existing methods.

The remainder of this paper is organized as follows. Section II presents related work. Section III gives background on LTL, SGs, and preliminary results on average cost per stage and average cost per cycle problem. Section IV introduces the system model. Section V presents the problem formulation on maximizing the probability of satisfying a given LTL specification and the corresponding solution algorithm. Section VI presents the problem formulation and solution algorithm of the problem of minimizing the average cost incurred due to violating invariant property while maximizing the probability of satisfying an LTL specification. Two numerical case studies are presented in Section VII to demonstrate our proposed approaches. Section VIII concludes the paper.

II. Related Work

Temporal logics such as LTL and CTL are widely used to specify and verify system properties [6], especially complex system behaviors. Multiple frameworks (e.g., receding horizon based [3], sampling based [4], sensor-based [19]–[21], probabilistic map based [22], multi-agent based [5], and probabilistic satisfaction based [23]) have been proposed for motion planning in robotics under temporal logic constraints. Control synthesis for deterministic system and probabilistic system under LTL formulas are studied in [24] and [9], respectively. Switching control policy synthesis among a set of shared autonomy systems is studied in [25]. When temporal logic constraints cannot be fulfilled [26], least-violating control synthesis problem is studied in [27]. These existing works do not consider the impact of malicious attacks.

Existing approaches of control synthesis under LTL constraints require a compact abstraction of CPS such as MDP [2], [8], [9], which models the non-determinism and probabilistic behaviors of the systems, and enables us applying off-the-shelf model checking algorithms for temporal logic [6]. Robust control of MDP under uncertainties has been extensively studied [8], [28]. Synthesis of control and sensing strategies under incomplete information for turn-based deterministic game is studied in [29]. However, MDPs only model the uncertainties that arise due to environmental disturbances and modeling errors, and are only suitable for scenarios with a single controller.

For CPS operating in adversarial scenarios, there are two decision makers (the controller and adversary) and their decisions are normally coupled. Thus, MDP cannot model the system, and the robust control strategy obtained on MDP may be suboptimal to the CPS operated in adversarial environment. To better formulate the strategic interactions between the controller and adversary, SG is used to generalize MDP [30]. Turn-based two-player SG, in which a unique player takes action at each time step, have been used to construct model checkers [17] and abstraction-refinement framework for model checking [7], [31], [32]. Unlike the literature using turn-based games [7], [17], [31], [32], however, we consider a different information structure denoted as concurrent SG, in which both players take actions simultaneously at each system state [33].

Several existing works using SGs focus on characterizing and computing Nash equilibria [34], [35], whereas in the present paper we consider a Stackelberg setting in which the adversary chooses an attack strategy based on the control policy selected by the system. The relationship between Nash and Stackelberg equilibria is investigated in [36]. A hybrid SG with asymmetric information structure is considered in [18]. The problem setting in [18] is similar to turn-based stochastic game, while the concurrent setting considered in this paper can potentially grants advantage to the controller (see [13] for a simple example). Moreover, the problem setting in this paper leads to a more general class of control strategies. Specifically, mixed strategies are considered in this work. In particular, concurrent SGs played with mixed strategies generalizes models including Markov chains, MDPs, probabilistic turn-based games and deterministic concurrent games [33]. The problem of maximizing the probability of satisfying a given specification consisting of safety and liveness constraints in the presence of adversary is considered in the preliminary conference version of this work [37]. Whereas only a restricted class of LTL specifications is considered in [37], in this paper, we derive results for arbitrary LTL specifications and we also investigate the problem of minimizing the rate of violating

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2019.2930039, IEEE Transactions on Automatic Control

invariant constraints.

CPS security is also investigated using game and control theoretic approaches. Secure state estimation is investigated in [38], [39]. CPS security and privacy using game theoretic approach is surveyed in [40]. Game theory based resilient control is considered in [41]. CPS security under Stackelberg setting and Nash setting are studied in [14] and [42], respectively. Stochastic Stackelberg security games have been studied in [15], [16].

III. Preliminaries

In this section, we present background on LTL, stochastic games, and preliminary results on the average cost per stage (ACPS) and average cost per cycle (ACPC) problems. Throughout this paper, we assume that inequalities between vectors and matrices are component wise comparison.

A. Linear Temporal Logic (LTL)

An LTL formula consists of [6]

- a set of atomic propositions Π;
- Boolean operators: negation ($\neg$), conjunction ($\wedge$) and disjunction ($\vee$).;
- temporal operators: next (X) and until ($\mathcal{U}$).

An LTL formula is defined inductively as

$$\phi = True \mid \pi \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid X\phi \mid \phi_1 \ \mathcal{U} \ \phi_2.$$

In other words, any atomic proposition ϕ is an LTL formula. Any formula formed by joining atomic propositions using Boolean or temporal connectives is an LTL formula. Other operators can be defined accordingly. In particular, implication ($\implies$) operator ($\phi \implies \psi$) can be described as $\neg\phi \vee \psi$; eventually ($\diamond$) operator $\diamond\phi$ can be written as $\diamond\phi = True \ \mathcal{U} \ \phi$; always ($\Box$) operator $\Box\phi$ can be represented as $\Box\phi = \neg\diamond\neg\phi$.

The semantics of LTL formulas are defined over infinite words in 2^{Π} [6]. Informally speaking, ϕ is true if and only if ϕ is true at the current time step. $\psi \ \mathcal{U} \ \phi$ is true if and only if $\psi \wedge \neg\phi$ is true until ϕ becomes true at some future time step. $\Box\phi$ is true if and only if ϕ is true for the current time step and all the future time. $\diamond\phi$ is true if ϕ is true at some future time. $X\phi$ is true if and only if ϕ is true in the next time step. A word η satisfying an LTL formula ϕ is denoted as $\eta \models \phi$.

Given any LTL formula, a deterministic Rabin automaton (DRA) can be constructed to represent the formula. A DRA is defined as follows.

Definition 1. *(Deterministic Rabin Automaton): A deterministic Rabin automaton (DRA) is a tuple $\mathcal{R} = (Q, \Sigma, \delta, q_0, Acc)$, where Q is a finite set of states, Σ is a finite set of symbols called alphabet, $\delta : Q \times \Sigma \to Q$ is the transition function, q_0 is the initial state and $Acc = \{(L(1), K(1)), (L(2), K(2)), \cdots, (L(Z), K(Z))\}$ is a finite set of Rabin pairs such that $L(z), K(z) \subseteq Q$ for all $z = 1, 2, \cdots, Z$ with Z being a positive integer.*

A run ρ of a DRA over a finite input word $\eta = \eta_0\eta_1\cdots\eta_n$ is a sequence of states $q_0q_1\cdots q_n$ such that $(q_{k-1}, \eta_k, q_k) \in \delta$ for all $0 \leq k \leq n$. A run ρ is accepted if and only if there exists a pair $(L(z), K(z))$ such that ρ intersects with $L(z)$ finitely many times and intersects with $K(z)$ infinitely often. Denote the satisfaction of a formula ϕ by a run ρ as $\rho \models \phi$.

B. Stochastic Games

A stochastic game is defined as follows [30].

Definition 2. *(Stochastic Game): A stochastic game (SG) $\mathcal{SG}$ is a tuple $\mathcal{SG} = (S, U_C, U_A, Pr, s_0, \Pi, \mathcal{L})$, where S is a finite set of states, U_C is a finite set of actions of the controller, U_A is a finite set of actions of an adversary, $Pr : S \times U_C \times U_A \times S \to [0, 1]$ is a transition function where $Pr(s, u_C, u_A, s')$ is the probability of a transition from state s to state s' when the controller's action is u_C and the adversary's action is u_A. $s_0 \in S$ is the initial state. Π is a set of atomic propositions. $\mathcal{L} : S \to 2^{\Pi}$ is a labeling function, which maps each state to a subset of propositions that are true at each state.*

Denote the set of admissible actions for the controller and adversary at state s as $U_C(s)$ and $U_A(s)$, respectively. Given a finite set S, we use the Kleene star S^* and the ω symbol S^ω to denote the set obtained by concatenating elements from S finitely and infinitely many times, respectively. Given an SG, the set of finite paths, i.e, the set of finite sequence of states, can be represented as S^*, while the set of infinite paths, i.e., the set of infinite sequence of states, can be represented as S^ω. The strategies (or policies) that players can commit to can be classified into the following two categories.

- *Pure strategy*: A pure strategy gives the action of the player as a deterministic function of the state. Suppose the players commit to pure strategies. Then a pure control strategy is defined as $\mu : S^* \to U_C$, which gives a specific control action, and a pure adversary strategy is defined as $\tau : S^* \to U_A$.
- *Mixed strategy*: A mixed strategy determines a probability distribution over all admissible pure strategies. Suppose the players commit to mixed strategy. Then a control policy for the controller is defined as $\mu : S^* \times U_C \to [0, 1]$, which maps a finite path and the admissible action to a probability distribution over the set of actions $U_C(s_k)$ available at state s_k. A policy τ for the adversary is defined as $\tau : S^* \times U_A \to [0, 1]$.

In this paper we focus on computing the optimal mixed strategy. When a specific action is assigned with probability one, then mixed strategy reduces to pure strategy. A control policy is *stationary* if it is only a function of the current state, i.e., $\mu : S \times U_C \to [0, 1]$ is only dependent on the last state of the path. A stationary policy is said to be *proper* if the probability of satisfying the given specification after finite steps is positive under this policy. Given a pair of policies μ and τ, an SG reduces to a Markov chain (MC) whose state set is S and transition probability from state s to s' is $P^{\mu\tau}(s, s') \triangleq \sum_{u_C \in U_C(s)} \sum_{u_A \in U_A(s)} \mu(s, u_C)\tau(s, u_A)Pr(s, u_C, u_A, s')$. Given a path $\beta \in S^\omega$, a word is generated as $\eta_\beta = \mathcal{L}(s_0)\mathcal{L}(s_1)\cdots$. The probability of satisfying an LTL formula ϕ under policies μ and τ on $\mathcal{SG}$ is denoted as $Pr^{\mu\tau}_{\mathcal{SG}} = Pr\{\eta_\beta \models \phi : \beta \in S^\omega\}$.

In the following, we review a subclass of stochastic games, denoted as Stackelberg games, involving two players [30]. In

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2019.2930039, IEEE Transactions on Automatic Control

the Stackelberg setting, player 1 (also called leader) commits to a strategy first. Then player 2 (also known as follower) observes the strategy of the leader and plays its best response. The information structure under Stackelberg setting can be classified into the following two categories.

- *Turn-based games*: Exactly one player is allowed to take action at each time step. Turn-based games are used to model asynchronous interaction between players.
- *Concurrent games*: All the players take actions simultaneously at each time step. Concurrent games are used to model synchronous interaction between players.

Unlike [7], [17], [31], [32], which are turn-based and played with pure strategies, in this paper, we focus on concurrent games with players committing to mixed strategies. To demonstrate the efficiency of mixed strategies in a concurrent game, we consider a robot moving to the right in a 1D space. The action sets for the controller and adversary are $\{move, stay\}$. If the controller and action take the same action at a given time step, then the robot will follow the specified action, i.e., move one step to the right under the action pair $(move, move)$ and stay in the current location under action pair $(stay, stay)$. The goal of the robot is to move to the location immediately to the right of its starting location. When the controller commits to a pure strategy, say $move$, then the adversary will always take action $stay$, and the robot will remain at its starting location. On the other hand, if the controller plays a mixed strategy, e.g., choosing $move$ and $stay$ with equal probability $1/2$ at each time step, then the robot has a $1/2$ probability to reach the desired location at each time step, and hence will reach the desired location within finite time with probability 1. On the other hand, in a turn-based setting where the adversary observes the controller's action before choosing its action at each time step, the adversary will always be able to choose the opposite of the controller's action and prevent the robot from moving. Hence while mixed strategies are beneficial in the concurrent game formulation, they are not beneficial in the turn-based game for this case.

The concept of Stackelberg equilibrium is used to solve Stackelberg games. The Stackelberg equilibrium is defined formally in the following.

Definition 3. *(Stackelberg Equilibrium): Denote the utility that the leader gains in a stochastic game $\mathcal{SG}$ under leader follower strategy pair (μ, τ) and the utility that the follower gains as $\mathcal{Q}_L(\mu, \tau)$ and $\mathcal{Q}_F(\mu, \tau)$, respectively. A pair of leader follower strategy (μ, τ) is a Stackelberg equilibrium if leader's strategy μ is optimal given that the follower observes its strategy and plays its best response, i.e., $\mu = \text{argmax}_{\mu' \in \boldsymbol{\mu}} \mathcal{Q}_L(\mu', \mathcal{BR}(\mu'))$, where $\boldsymbol{\mu}$ is the set of all admissible policies of the controller and $\mathcal{BR}(\mu') = \{\tau : \tau = \text{argmax}\, \mathcal{Q}_F(\mu', \tau)\}$ is the best response to leader's strategy μ' played by the follower.*

C. ACPS and ACPC Problems

We present some preliminary results on the average cost per stage (ACPS) problem and average cost per cycle (ACPC) problem on MDP without the presence of adversary in this subsection. Both problems focus on deterministic control policies $\mu : S \to U_C$ and unichain MDP. An MDP is said to be unichain if for any control policy μ, the induced MC is irreducible, i.e, the probability of reaching any state from any state on the MC is positive. Denote the cost incurred at state s by applying the deterministic control policy μ as $g(s, \mu(s))$. The transition probability from state s to s' via action u on MDP is denoted as $Pr(s, u, s')$. Each transition to a new state is viewed as a completion of a stage. Then the objective of ACPS problem is to minimize

$$J_\mu(s) = \limsup_{N\to\infty} \frac{1}{N} \mathbb{E}\left\{ \sum_{k=0}^{N} g(s_k, \mu(s_k)) | s_0 = s \right\} \quad (1)$$

over all deterministic stationary control policies.

It has been shown that a gain-bias pair (J_μ, h_μ) for ACPS problem satisfies the properties stated as follows.

Proposition 1 ([43]). *Assume the MDP is unichain. Then:*

- *the optimal ACPS $J^*_\mu(s_0)$ associated with each control policy μ is independent of initial state s_0, i.e., there exists a constant J^*_μ such that $J^*_\mu(s_0) = J^*_\mu$ for all $s_0 \in S$;*
- *there exists a vector h such that $J^*_\mu + h_\mu(s) = \min_\mu \left\{ g(s, \mu(s)) + \sum_{s' \in S} Pr(s, \mu(s), s') h(s') \right\}$.*

We present some preliminary results on the ACPC problem on MDP in the following. Denote the set of states that satisfy LTL formula ϕ as S_ϕ. A cycle is completed when S_ϕ is visited. Therefore, a path starting from s_0 and ending in S_ϕ completes the first cycle, and the path starting from S_ϕ after completing the first cycle completes the second cycle when coming back to S_ϕ. Denote the number of cycles that have been completed until stage N as $C(N)$. The ACPC problem is described as given an MDP and an LTL formula ϕ, find a control policy μ that minimizes

$$J_\mu(s_0) = \limsup_{N\to\infty} \mathbb{E}\left\{ \frac{\sum_{k=0}^{N} g(s_k, \mu(s_k))}{C(N)} \Big| \eta_\mu \models \phi \right\}, \quad (2)$$

where $\eta_\mu = \mathcal{L}(s_0)\mathcal{L}(s_1)\cdots$ is the word generated by the path $s_0 s_1 \cdots$ induced by deterministic control policy μ.

It has been shown that the following proposition holds for the ACPC problem.

Proposition 2 ([9]). *Assume the MDP is unichain. Then:*

- *the optimal ACPC $J^*_\mu(s_0)$ associated with each control policy μ is independent of initial state s_0, i.e., there exists a constant J^*_μ such that $J^*_\mu(s_0) = J^*_\mu$ for all $s_0 \in S$;*
- *there exists some vector h such that the following equation holds $J^*_\mu + h(s) = \min_\mu \Big\{ g(s, \mu(s)) + \sum_{s' \in S} Pr(s, \mu(s), s') h(s') + J^*_\mu \sum_{s' \in S} Pr(s, \mu(s), s') \Big\}$.*

IV. System Model

In this section, we present the system model. We consider the following discrete-time finite state system

$$x(t+1) = f(x(t), u_C(t), u_A(t), \vartheta(t)), \ \forall t = 0, 1, \cdots, \quad (3)$$

where $x(t)$ is the finite system state, $u_C(t)$ is the control input from the controller, $u_A(t)$ is the attack signal from the adversary, and $\vartheta(t)$ is stochastic disturbance.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2019.2930039, IEEE Transactions on Automatic Control

In system (3), there exists a strategic adversary that can tamper with the system transition. In particular, the controller and adversary jointly determine the state transition. For instance, an adversary that launches false data injection attack modifies the control input as $u(t) = u_C(t) + u_A(t)$; an adversary that launches denial-of-service attack manipulates the control input as $u(t) = u_C(t) \cdot u_A(t)$, where $u_A(t) \in \{0, 1\}$.

In security domain, Stackelberg game is widely used to model systems in the presence of malicious attackers. In Stackelberg setting, the controller plays as the leader and the adversary plays as the follower. In this paper, we adopt the concurrent Stackelberg setting. The controller first commits to its control strategy. The adversary can stay outside for indefinitely long time to observe the strategy of the controller and then chooses its best response to the controller's strategy. However, at each time step, both players must take actions simultaneously. The system is given some specification that is modeled using LTL.

To abstract system (3) as a finite state/action SG, we propose a heuristic simulation based algorithm as shown in Algorithm 1, which is generalized from the approaches proposed in [8], [44]. The difference between Algorithm 1 and algorithms in [8], [44] is that Algorithm 1 considers the presence of adversary. Algorithm 1 takes the dynamical system (3), the set of sub-regions of state space $\{X_1, \cdots, X_n\}$ and actions as inputs. We observe that the choice of subregions $X_1, \ldots, X_n$ may affect the accuracy of the model, however, choice of the subregions is beyond the scope of this work. For each sub-region X_i and pair of (control, adversary) inputs (u_C, u_A), we randomly select K sample states in X_i and adversary and control inputs that map to u_C and u_A. We compute the probability distribution over the set of sub-regions $\{X_j\}$ that the system can transition to following (3), and update $Pr(X_i, u_C, u_A, X_j)$ accordingly for all X_j (Algorithm 1). To approximate the transition probability, Monte Carlo simulation or particle filter can be used [8], [44], [45].

Algorithm 1 Algorithm for constructing a stochastic game approximation of a system.

procedure CREATE_STOCHASTIC_GAME($X_1, \ldots, X_n$)
  Input: Dynamics (3), set of subsets $X_1, \ldots, X_n$
  Output: Stochastic game $\mathcal{SG} = (S, U_C, U_A, Pr, s_0, \Pi, \mathcal{L})$
  Initialize K
  $S = \{X_1, \ldots, X_n\}$ and $\mathcal{L}$ is determined accordingly
  Generate control primitive sets $U_C = \{u_{C_1}, u_{C_2} \cdots, u_{C_\Xi}\}$ and $U_A = \{u_{A_1}, u_{A_2} \cdots, u_{A_\Gamma}\}$
  for $i = 1, \ldots, n$ **do**
    for all $u_C \in U_C$ and $u_A \in U_A$ **do**
      for $k = 1, \ldots, K$ **do**
        $x \leftarrow$ sampled state in X_i
        $\hat{u}_C, \hat{u}_A \leftarrow$ sampled inputs from u_C, u_A
        $j \leftarrow$ region containing $f(x, \hat{u}_C, \hat{u}_A, \vartheta)$
        Invoke particle filter to approximate transition probabilities Pr between sub-region i and j for all i and j.
      end for
    end for
  end for
end procedure

In the following, we present two applications in the security domain that can be formulated using the proposed framework.

1) Infrastructure Protection in Power System: The proposed framework can capture attack-defense problems on power system as shown in the following. An attack-defense problem on power system is investigated in [34].

The players involved in this example are the power system administrator and adversary. The adversary aims to disrupt the transmission lines in power network, while the administrator deploys resources to protect critical infrastructures or repair damaged infrastructures. The dynamics of power system is modeled as $x(t+1) = f(x(t), u_C(t), u_A(t))$, where $x(t)$ is the state vector, $u_C(t)$ and $u_A(t)$ are the inputs from the administrator and adversary, respectively. Depending on the focus of the administrator, the state may contain bus voltages, bus power injections, network frequency, and so on. The actions of the administrator U_C and adversary U_A, respectively, are the actions to protect (by deploying protection or repair resources) and damage (by opening the breakers at ends of) the transmission lines. If an attack is successful, then the transmission line is out of service, which will result in dramatic change on state vector. Thus the states evolve following the joint actions of administrator and adversary. Moreover, the probability of the occurrence of events, i.e., the transmission line is out of service, is jointly determined by the actions of adversary and administrator (or defender). The specifications that can be given to the system might include reachability (e.g., 'eventually satisfy optimal power flow equation': $\diamond OPF$) and reactivity (e.g., 'if voltage exceeds some threshold, request load shedding from demand side': $\square(voltage\_alarm \implies X\ DR)$).

2) Networked Control System under Attacks: In the following, we present an example on control synthesis for networked control system under deception attacks.

The system is modeled as a discrete linear time invariant system $x(k+1) = Ax(k) + Bu(k) + \vartheta(k),\ k = 0, 1 \cdots$, where $x(k)$ is the system state, $u(k)$ is the compromised control input and $\vartheta(k)$ is independent Gaussian distributed disturbance. There exists an intelligent and strategic adversary that can compromise the control input of the system by launching deception attack. When the adversary launches deception attack on system actuator [14], then the control input is represented as $u(k) = u_C(k) + u_A(k)$. Typical specifications that are assigned to the system include stability and safety (e.g., 'eventually reach stable status while not reaching unsafe state': $\diamond\square$stable $\wedge\ \square\neg$unsafe).

V. Problem Formulation - Maximizing Satisfaction Probability

In this section, we formulate the problem of maximizing the probability of satisfying a given LTL specification in the presence of an adversary. We first present the problem formulation, and then give a solution algorithm for computing the optimal control policy.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2019.2930039, IEEE Transactions on Automatic Control

A. Problem Statement

The problem formulation is as follows.

Problem 1. *Given a stochastic game $\mathcal{SG}$ and an LTL specification ϕ, compute a control policy μ that maximizes the probability of satisfying the specification ϕ under any adversary policy τ, i.e.,*

$$\max_{\mu} \min_{\tau} Pr^{\mu\tau}_{\mathcal{SG}}(\phi). \tag{4}$$

Denote the probability of satisfying specification ϕ as satisfaction probability. The policies μ and τ that achieve the max-min value of (4) can be interpreted as an equilibrium defined in Definition 3 in a zero-sum Stackelberg game between the controller and adversary, in which the controller first chooses a randomized policy μ, and the adversary observes μ and selects a policy τ to minimize $Pr^{\mu\tau}_{\mathcal{SG}}(\phi|s)$. By Von Neumann's theorem [46], the satisfaction probability at equilibrium must exist. We restrict our attention to the class of stationary policies, leaving the general case for future work. We have the following preliminary lemma.

Lemma 1. *Let satisfaction probability $v(s) = \max_{\mu}\min_{\tau} Pr^{\mu\tau}_{\mathcal{SG}}(\phi|s)$. Then*

$$v(s) = \max_{\mu}\min_{\tau} \sum_{u_C\in U_C(s)} \sum_{u_A\in U_A(s)} \sum_{s'\in S} \mu(s,u_C) \tau(s,u_A) v(s') Pr(s,u_C,u_A,s'). \tag{5}$$

Conversely, if $v(s)$ satisfies (5), then $v(s) = \max_{\mu}\min_{\tau} Pr^{\mu\tau}_{\mathcal{SG}}(\phi|s)$. Moreover, the satisfaction probability v is unique.

Proof. In the following, we will first show the forward direction. We let $n = |S|$ and define three operators $T_{\mu\tau} : [0,1]^n \to [0,1]^n$, $T_\mu : [0,1]^n \to [0,1]^n$, and $T : [0,1]^n \to [0,1]^n$.

$$(T_{\mu\tau}v)(s) = \sum_{s'} Pr(s,\mu,\tau,s')v(s')$$
$$(T_{\mu}v)(s) = \min_{\tau} \sum_{s'} Pr(s,\mu,\tau,s')v(s')$$
$$(Tv)(s) = \max_{\mu}\min_{\tau} \sum_{s'} Pr(s,\mu,\tau,s')v(s')$$

where $Pr(s,\mu,\tau,s') = \sum_{u_C\in U_C(s)} \sum_{u_A\in U_A(s)} \mu(s,u_C) \tau(s,u_A) Pr(s,u_C,u_A,s')$. Suppose that μ is a Stackelberg equilibrium with $v(s)$ equal to the satisfaction probability for state s, and yet (5) does not hold. We have that $v = T_\mu v$, since v is the optimal policy for the MDP defined by the policy μ [43]. On the other hand, $T_\mu v \le Tv$. Composing T and T_μ k times and taking the limit as k tends to infinity yields $v = \lim_{k\to\infty} T^k_\mu v \le \lim_{k\to\infty} T^k v \triangleq v^*$. The convergence of $T^k v$ to a fixed point v^* follows from the fact that T is a bounded and monotone nondecreasing operator. Furthermore, choosing the policy $\mu(s)$ at each state as the maximizer of (5) yields a policy with satisfaction probability v^*. Hence $v \le v^*$. If $v(s) = v^*(s)$ for all states s, then (5) is satisfied, contradicting the assumption that the equation does not hold. On the other hand, if $v(s) < v^*(s)$ for some state s, then μ is not a Stackelberg equilibrium.

We next show that the vector v satisfying (5) is unique. Since every Stackelberg equilibrium satisfies (5), if the vector v is unique, then the vector v must be a Stackelberg equilibrium. Suppose that uniqueness does not hold, and let μ and μ' be Stackelberg equilibrium policies with corresponding satisfaction probabilities v and v'. We have that $v = Tv \ge T_{\mu'}v$. Composing k times and taking the limit as k tends to infinity, we have $v = \lim_{k\to\infty} T^k v \ge \lim_{k\to\infty} T^k_{\mu'} v = v'$. By the same argument, $v' \ge v$, implying that $v = v'$ and thus uniqueness holds. □

By Lemma 1, we have that the satisfaction probability for some state s can be computed as the linear combination of the satisfaction probabilities of its neighbor states, where the coefficients are the transition probabilities jointly determined by the control and adversary policies. Lemma 1 provides us the potential to apply iterative algorithm to compute the satisfaction probability.

B. Computing the Optimal Policy

Motivated by model checking algorithms [6], we first construct a product SG. Then we analyze Problem 1 on the product SG. A product SG is defined as follows.

Definition 4. (Product SG): *Given an SG $\mathcal{SG} = (S, U_C, U_A, Pr, \mathcal{L}, \Pi)$ and a DRA $\mathcal{R} = (Q, \Sigma, \delta, q_0, Acc)$, a (labeled) product SG is a tuple $\mathcal{G} = (S_\mathcal{G}, U_C, U_A, Pr_\mathcal{G}, Acc_\mathcal{G})$, where $S_\mathcal{G} = S \times Q$ is a finite set of states, U_C is a finite set of control inputs, U_A is a finite set of attack signals, $Pr_\mathcal{G}((s,q),u_C,u_A,(s',q')) = Pr(s,u_C,u_A,s')$ if $\delta(q,\mathcal{L}(s')) = q'$, $Acc_\mathcal{G} = \{(L_\mathcal{G}(1), K_\mathcal{G}(1)), (L_\mathcal{G}(2), K_\mathcal{G}(2)), \cdots, (L_\mathcal{G}(Z), K_\mathcal{G}(Z))\}$ is a finite set of Rabin pairs such that $L_\mathcal{G}(z), K_\mathcal{G}(z) \subseteq S_\mathcal{G}$ for all $z = 1, 2, \cdots, Z$ with Z being a positive integer. In particular, a state $(s,q) \in L_\mathcal{G}(z)$ if and only if $q \in L(z)$, and a state $(s,q) \in K_\mathcal{G}(z)$ if and only if $q \in K(z)$.*

By Definition 2 and Definition 4, we have the following observations. First, since the transition probability is determined by $\mathcal{SG}$ and the satisfaction condition is determined by $\mathcal{R}$, the satisfaction probability of ϕ on $\mathcal{SG}$ is equal to the satisfaction probability of ϕ on the product SG $\mathcal{G}$. Second, we can generate the corresponding path $s_0 s_1 \cdots$ on $\mathcal{SG}$ given a path $(s_0,q_0)(s_1,q_1)\cdots$ on the product SG $\mathcal{G}$. Finally, given a control policy μ synthesized on the product SG $\mathcal{G}$, a corresponding control policy $\mu_\mathcal{SG}$ on $\mathcal{SG}$ is obtained by letting $\mu_\mathcal{SG}(s_i) = \mu((s_i,q))$ for all time step i [6], [8]. Due to these one-to-one correspondence relationships, in the following, we analyze Problem 1 on the product SG $\mathcal{G}$ and present an algorithm to compute the optimal control policy. When the context is clear, we use s to represent state $(s,q) \in S_\mathcal{G}$.

We next introduce the concept of Generalized Accepting Maximal End Component (GAMEC), which is generalized from accepting maximal end component (AMEC) on MDP.

Definition 5. (Sub-SG): *A sub-SG of an SG $\mathcal{SG} = (S, U_C, U_A, Pr, s_0, \Pi, \mathcal{L})$ is a pair of states and actions (C,D) where $\emptyset \ne C \subseteq S$ is a set of states, and $D : C \to 2^{U_C(s)}$ is an enabling function such that $D(s) \subseteq U_C(s)$ for*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2019.2930039, IEEE Transactions on Automatic Control

all $s \in C$ *and* $\{s' | Pr(s, u_C, u_A, s') > 0, \forall u_A \in U_A(s), s \in C\} \subseteq C$.

By Definition 5, we have a sub-SG is also an SG. Given Definition 5, a Generalized Maximal End Component (GMEC) is defined as follows.

Definition 6. *A Generalized End Component (GEC) is a sub-SG* (C, D) *such that the underlying digraph* $G_{(C,D)}$ *of sub-SG* (C, D) *is strongly connected. A GMEC is a GEC* (C, D) *such that there exists no other GEC* $(C', D') \neq (C, D)$*, where* $C \subseteq C'$ *and* $D(s) \subseteq D'(s)$ *for all* $s \in C$.

Definition 7. *A GAMEC on the product SG* $\mathcal{G}$ *is a GMEC if there exists some* $(L_{\mathcal{G}}(z), K_{\mathcal{G}}(z)) \in Acc_{\mathcal{G}}$ *such that* $L_{\mathcal{G}}(z) \cap C = \emptyset$ *and* $K_{\mathcal{G}}(z) \subseteq C$.

By Definition 7, we have a set of states constitutes a GAMEC if there exists a control policy such that for any initial states in the GAMEC, the system remains in the GAMEC with probability one and the specification is satisfied with probability one. We denote the set of GAMECs as $\mathcal{C}$, and the set of states that constitute GAMEC as accepting states. Algorithm 2 is used to compute the set of GAMECs. Given a product SG $\mathcal{G}$, a set of GAMECs $\mathcal{C}$ can be initialized as $C = \emptyset$ and $D(s) = U_C(s)$ for all s. Also, we define a temporary set $\mathcal{C}_{temp}$ which is initialized as $\mathcal{C}_{temp} = S_{\mathcal{G}}$. Then from line 8 to line 17, we compute a set of states R that should be removed from GMEC. The set R is first initialized to be empty. Then for each state s in each nontrivial strongly connected component (SCC) of the underlying diagraph, i.e., the SCC with more than one states, we modify the admissible actions at state s by keeping the actions that can make the system remain in C under any adversary action. If there exists no such admissible action at state s, then the state s is added into R. From line 18 to line 26, we examine if there exists any state s' in current GMEC that will steer the system into states in R. In particular, by taking action u_C at each state s', if there exists some adversary action u_A such that the system is steered into some state $s \in R$, then u_C is removed from $U_C(s')$. Moreover, if there exists no admissible action at state s', then s' is added to R. Then we update the GMEC set as shown from line 27 to line 32. This procedure is repeated until no further update can be made on GMEC set. Line 34 to line 40 is to find the GAMEC following Definition 7. Given the set of GAMECs $\mathcal{C} = \{(C_1, D_1), \cdots, (C_h, D_h), \cdots, (C_{|\mathcal{C}|}, D_{|\mathcal{C}|})\}$ returned by Algorithm 2, the set of accepting states $\mathcal{E}$ is computed as $\mathcal{E} = \cup_{h=1}^{|\mathcal{C}|} C_h$.

The main idea to computing the solution to (4) is to show that the max-min probability of (4) is equivalent to maximizing (over μ) the worst-case probability of reaching the set of accepting states $\mathcal{E}$. Denote the probability of reaching the set of accepting states $\mathcal{E}$ as reachability probability. In the following, we formally prove the equivalence between the worst-case satisfaction probability of (4) and the worst-case reachability probability. Then, we present an efficient algorithm for computing a policy μ that maximizes the worst-case probability of reaching $\mathcal{E}$, with the proofs of the correctness and convergence of the proposed algorithm. In particular, our proposed solution is based on the following.

Algorithm 2 Computing the set of GAMECs $\mathcal{C}$.

1: **procedure** COMPUTE_GAMEC($\mathcal{G}$)
2: **Input**: Product SG $\mathcal{G}$
3: **Output:** Set of GAMECs $\mathcal{C}$
4: **Initialization:** Let $D(s) = U_C(s)$ for all $s \in S_{\mathcal{G}}$. Let $\mathcal{C} = \emptyset$ and $\mathcal{C}_{temp} = \{S_{\mathcal{G}}\}$
5: **repeat**
6: $\mathcal{C} = \mathcal{C}_{temp}$, $\mathcal{C}_{temp} = \emptyset$
7: **for** $C \in \mathcal{C}$ **do**
8: $R = \emptyset$ ▷ R is the set of states that should be removed
9: Let $SCC_1, \cdots, SCC_n$ be the set of nontrivial strongly connected components (SCC) of the underlying diagraph $G_{(C,D)}$
10: **for** $i = 1, \cdots, n$ **do**
11: **for** each state $s \in SCC_i$ **do**
12: $D(s) = \{u_C \in U_C(s) | s' \in C$ where $Pr(s, u_C, u_A, s') > 0, \ \forall u_A \in U_A(s)\}$
13: **if** $D(s) = \emptyset$ **then**
14: $R = R \cup \{s\}$
15: **end if**
16: **end for**
17: **end for**
18: **while** $R \neq \emptyset$ **do**
19: dequeue $s \in R$ from R and C
20: **if** there exist $s' \in C$ and $u_C \in U_C(s')$ such that $Pr(s', u_C, u_A, s) > 0$ under some $u_A \in U_A(s')$ **then**
21: $D(s') = D(s') \setminus \{u_C\}$
22: **if** $D(s') = \emptyset$ **then**
23: $R = R \cup \{s'\}$
24: **end if**
25: **end if**
26: **end while**
27: **for** $i = 1, \cdots, n$ **do**
28: **if** $C \cap SCC_i \neq \emptyset$ **then**
29: $\mathcal{C} = \mathcal{C}_{temp} \cup \{C \cap SCC_i\}$
30: **end if**
31: **end for**
32: **end for**
33: **until** $\mathcal{C} = \mathcal{C}_{temp}$
34: **for** $C \in \mathcal{C}$ **do**
35: **for** $(L_{\mathcal{G}}(z), K_{\mathcal{G}}(z)) \in \text{Acc}_{\mathcal{G}}$ **do**
36: **if** $L_{\mathcal{G}}(z) \cap C \neq \emptyset$ or $K_{\mathcal{G}}(z) \not\subseteq C$ **then**
37: $\mathcal{C} = \mathcal{C} \setminus C$
38: **end if**
39: **end for**
40: **end for**
41: **return** $\mathcal{C}$
42: **end procedure**

Proposition 3. *For any stationary control policy* μ *and initial state* s*, the minimum probability over all stationary adversary policies of satisfying the LTL formula is equal to the minimum probability over all stationary policies of reaching* $\mathcal{E}$*, i.e., given any stationary policy* μ*, we have*

$$\min_{\tau} Pr_{\mathcal{G}}^{\mu\tau}(\phi | s) = \min_{\tau} Pr_{\mathcal{G}}^{\mu\tau}(\textit{reach } \mathcal{E} | s), \tag{6}$$

where $Pr_{\mathcal{G}}^{\mu\tau}(\textit{reach } \mathcal{E})$ *is the probability of reaching* $\mathcal{E}$ *under policies* μ *and* τ.

Proof. By Definition of $\mathcal{E}$, if the system reaches $\mathcal{E}$, then ϕ is satisfied for a maximizing policy μ. Thus $\min_\tau Pr_{\mathcal{G}}^{\mu\tau}(\text{reach } \mathcal{E}) = \min_\tau Pr_{\mathcal{G}}^{\mu\tau}(\phi)$.

Suppose that for some control policy μ and initial state s_0,

$$\min_{\tau} Pr_{\mathcal{G}}^{\mu\tau}(\phi | s_0) > \min_{\tau} Pr_{\mathcal{G}}^{\mu\tau}(\text{reach } \mathcal{E} | s_0), \tag{7}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2019.2930039, IEEE Transactions on Automatic Control

and let τ be a minimizing stationary policy for the adversary. The policies μ and τ induce an MC on the state space. By model checking algorithms on MC [6], the probability of satisfying ϕ from s_0 is equal to the probability of reaching a bottom strongly connected component (BSCC) that satisfies ϕ. By assumption there exists a BSCC, denoted SCC_0, that is reachable from s_0, disjoint from $\mathcal{E}$, and yet satisfies $Pr_{\mathcal{G}}^{\mu\tau}(\phi|s) = 1$ for all $s \in S_0$ (if this were not the case, then (7) would not hold).

Choose a state $s \in SCC_0$. Since $s \notin \mathcal{E}$, there exists a policy $\hat{\tau}$ such that $Pr_{\mathcal{P}}^{\mu\hat{\tau}}(\phi|s) < 1$. Create a new adversary policy τ_1 as $\tau_1(s') = \hat{\tau}(s')$ for all $s' \in SCC_0$ and $\tau_1(s') = \tau(s')$ otherwise. This policy induces a new MC on the state space. Furthermore, since only the outgoing transitions from SCC_0 are affected, the success probabilities of all sample paths that do not reach SCC_0 are unchanged.

If there exists any state s' that is reachable from s in the new chain with $Pr_{\mathcal{G}}^{\mu\tau_1}(\phi|s') < 1$, then the policy τ_1 strictly reduces the probability of satisfying ϕ, thus contradicting the assumption that τ is a minimizing policy. Otherwise, let SCC_1 denote the set of states that are reachable from s under μ and τ_1 and are disjoint from $\mathcal{E}$ (this set must be non-empty; otherwise, the policy $\hat{\tau}$ would lead to $Pr_{\mathcal{G}}^{\mu\hat{\tau}}(\phi|s) = 1$, a contradiction). Construct a new policy τ_2 by $\tau_2(s') = \hat{\tau}(s')$ if $s' \in S_1$ and $\tau_2(s') = \tau(s')$ otherwise. Proceeding inductively, we derive a sequence of policies τ_k that satisfy $Pr_{\mathcal{G}}^{\mu\tau_k}(\phi) \leq Pr_{\mathcal{G}}^{\mu\tau}(\phi)$. This process terminates when either $Pr_{\mathcal{G}}^{\mu\tau_k}(\phi|s_0) < Pr_{\mathcal{G}}^{\mu\tau}(\phi|s_0)$, contradicting the minimality of τ, or when $Pr_{\mathcal{G}}^{\mu\tau_k}(\phi|s'') = Pr_{\mathcal{G}}^{\mu\hat{\tau}}(\phi|s'')$ for all s'' that are reachable from s under $\hat{\tau}$. The latter case, however, implies that $Pr_{\mathcal{G}}^{\mu\hat{\tau}}(\phi|s)) = 1$, contradicting the definition of $\hat{\tau}$. □

Proposition 3 implies that the problem of maximizing the worst-case success probability can be mapped to a reachability problem on the product SG $\mathcal{G}$, where $\mathcal{G}$ is modified following Algorithm 3. A dummy state $dest$ is added into the state space of $S_{\mathcal{G}}$. All transitions starting from a state in GAMECs are directed to state $dest$ with probability one regardless of the actions taken by the adversary. The transition probabilities and action spaces of all other nodes are unchanged. We observe that the reachability probability remains unchanged after applying Algorithm 3. Hence the satisfaction probability remains unchanged. Moreover, the one-to-one correspondence of control policy still holds for states outside $\mathcal{E}$. Therefore, Problem 1 is then equivalent to

$$\max_{\mu} \min_{\tau} Pr_{\mathcal{G}}^{\mu\tau}(\text{reach } dest) \tag{8}$$

Then, the solution to (4) can be obtained from the solution to (8) by following the optimal policy μ^* for (8) at all states not in $\mathcal{E}$. The control policy for states in $\mathcal{E}$ can be any probability distribution over the set of enabled actions in each GAMEC.

Due to Proposition 3, in the following we focus on solving the problem (8). Our approach for solving (8) is to first compute a value vector $v \in \mathbb{R}^{|S_{\mathcal{G}}|}$, where $v(s) = \max_{\mu} \min_{\tau} Pr_{\mathcal{G}}^{\mu\tau}(\text{reach } dest|s)$. By Lemma 1, the optimal policy can then be obtained from v by choosing the distribution μ that solves the optimization problem of (5) at each state s. Algorithm 4 gives a value iteration based algorithm for computing v. The idea of the algorithm is to initialize v to be zero except on states in $\mathcal{E}$, and then greedily update $v(s)$ at each iteration by computing the optimal Stackelberg policy at each state. The algorithm terminates when a stationary v is reached.

Algorithm 3 Modifying product SG $\mathcal{G}$.

procedure CONSTRUCT_SG($\mathcal{G}$, $\mathcal{C}$)
  Input: Product SG $\mathcal{G}$, the set of GAMECs $\mathcal{C}$
  Output: Modified product SG $\mathcal{G}$
  $S_{\mathcal{G}} := S_{\mathcal{G}} \cup \{dest\}$,$U_C(s) := U_C(s) \cup \{d\},\ \forall s \in S_{\mathcal{G}}$
  $Pr_{\mathcal{G}}(s, d, u_A, dest) = 1$ for all $s \in \mathcal{E} \cup \{dest\}$ and $u_A \in U_A(s)$
end procedure

Algorithm 4 Algorithm for a control strategy that maximizes the probability of satisfying ϕ.

procedure MAX_REACHABILITY($\mathcal{G}$, $\mathcal{C}$)
  Input: product SG $\mathcal{G}$, the set of GAMECs $\mathcal{C}$
  Output: Vector $v \in \mathbb{R}^{|S_{\mathcal{G}}|}$, where $v(s) = \max \min Pr_{\mathcal{G}}^{\mu\tau}(\text{reach } dest|s_0 = s)$
  Initialization: $v^0 \leftarrow 0$, $v^1(s) \leftarrow 1$ for $s \in \mathcal{E}$, $v^1(s) \leftarrow 0$ otherwise, $k \leftarrow 0$
  while $\max\{|v^{k+1}(s) - v^k(s)| : s \in S_{\mathcal{G}}\} > \delta$ **do**
    $k \leftarrow k + 1$
    for $s \notin \mathcal{E}$ **do**
      Compute v as $v^{k+1}(s) \leftarrow \max_{\mu} \min_{\tau} \Big\{ \sum_{s'} \sum_{u_C \in U_C(s)} \sum_{u_A \in U_A(s)} v(s')\mu(s, u_C)\tau(s, u_A) Pr_{\mathcal{G}}(s, u_C, u_A, s') \Big\}$
    end for
  end while
  return v
end procedure

The following theorem shows that Algorithm 4 guarantees convergence to a Stackelberg equilibrium.

Theorem 1. *There exists v^∞ such that for any $\epsilon > 0$, there exists δ and K such that $||v^k - v^\infty||_\infty < \epsilon$ for $k > K$. Furthermore, v^∞ satisfies the conditions of v in Lemma 1.*

Proof. We first show that, for each s, the sequence $v^k(s) : k = 1, 2, \ldots,$ is bounded and monotone. Boundedness follows from the fact that, at each iteration, $v^k(s)$ is a convex combination of the states of its neighbors, which are bounded above by 1. To show monotonicity, we induct on k. Note that $v^1(s) \geq v^0(s)$ and $v^2(s) \geq v^1(s)$ since $v^1(s) = 0$ for $s \notin \mathcal{E}$ and $v^k(s) \equiv 1$ for $s \in \mathcal{E}$.

Let μ^k denote the optimal control policy at step k. We have

$$\begin{aligned} v^{k+1}(s) &\geq \min_{\tau} \sum_{u_C \in U_C(s)} \sum_{u_A \in U_A(s)} \sum_{s' \in S} v^k(s') \\ &\quad \cdot \mu^k(s, u_C)\tau(s, u_A) Pr_{\mathcal{G}}(s, u_C, u_A, s') \qquad (9) \\ &\geq \min_{\tau} \sum_{u_C \in U_C(s)} \sum_{u_A \in U_A(s)} \sum_{s' \in S} v^{k-1}(s') \qquad (10) \end{aligned}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2019.2930039, IEEE Transactions on Automatic Control

$$\begin{aligned} & \cdot \mu^k(s,u_C)\tau(s,u_A)Pr_{\mathcal{G}}(s,u_C,u_A,s') \\ = \; & v^k(s) \end{aligned} \tag{11}$$

Eq. (9) follows because the value of $v^{k+1}(s)$, which corresponds to the maximizing policy, dominates the value achieved by the particular policy μ_s^k. Eq. (9) holds by induction, since $v^k(s') \geq v^{k-1}(s')$ for all s'. Finally, (11) holds by construction of μ_s^k. Hence $v^k(s)$ is monotone in k.

We therefore have that $v^k(s)$ is a bounded monotone sequence, and hence converges by the monotone convergence theorem. Let v^∞ denote the vector of limit points, so that we can select δ sufficiently small (to prevent the algorithm from terminating before convergence) and K large in order to satisfy $||v^k - v^\infty||_\infty < \epsilon$.

We now show that v^∞ is a Stackelberg equilibrium. Since $v^k(s)$ converges, it is a Cauchy sequence and thus for any $\epsilon > 0$, there exists K such that $k > K$ implies that $|v^k(s) - v^{k+1}(s)| < \epsilon$. By construction, this is equivalent to

$$\begin{aligned} \Big| v^k(s) - \max_{\mu} \min_{\tau} \sum_{u_C \in U_C(s)} \sum_{u_A \in U_A(s)} \sum_{s' \in S} \big[v^{k-1}(s')\mu(s,u_C) \\ \tau(s,u_A) Pr_{\mathcal{G}}(s,u_C,u_A,s') \big] \Big| < \epsilon, \end{aligned}$$

and hence v^∞ is within ϵ of a Stackelberg equilibrium for every $\epsilon > 0$. □

While this approach guarantees asymptotic convergence to a Stackelberg equilibrium, there is no guarantee on the rate of convergence. By modifying Line 8 of the algorithm so that $v^{k+1}(s)$ is updated if

$$\begin{aligned} \max_{\mu} \min_{\tau} \sum_{u_C \in U_C(s)} \sum_{u_A \in U_A(s)} \sum_{s' \in S} \Big[v(s')\mu(s,u_C)\tau(s,u_A) \\ Pr(s,u_C,u_A,s') \Big] > (1+\epsilon) v^k(s) \end{aligned} \tag{12}$$

and is constant otherwise, we derive the following result on the termination time.

Proposition 4. *The ϵ-relaxation of (12) converges to a value of v satisfying $\max\{|v^{k+1}(s) - v^k(s)| : s \in S_{\mathcal{G}}\} < \epsilon$ within $n \max_s \left\{ \log\left(\frac{1}{v^0(s)}\right) / \log(1+\epsilon) \right\}$ iterations, where $v^0(s)$ is the smallest positive value of $v^k(s)$ for $k = 0, 1, \ldots$.*

Proof. After N updates, we have that $v^N(s) \geq (1+\epsilon)^N v^0(s)$. Hence for each s, $v(s)$ will be incremented at most $\max_s \left\{ \log\left(\frac{1}{v^0(s)}\right) / \log(1+\epsilon) \right\}$ times. Furthermore, we have that at least one $v(s)$ must be updated at each iteration, thus giving the desired upper bound on the number of iterations. By definition of (12), the set that is returned satisfies $|v^{k+1}(s) - v^k(s)| < \epsilon v^k(s) < \epsilon$. □

VI. Problem Formulation-Minimizing Invariant Constraint Violation

In this section, we focus on a subclass of specifications of the form $\phi = \phi_1 \wedge \psi$, where ϕ_1 is an arbitrary LTL formula and ψ is an invariant constraint. An invariant constraint requires the system to always satisfy some property. The general LTL formula ϕ_1 can be used to model any arbitrary properties such as liveness $\phi_1 = \square\diamond\pi$, while the invariant property can be used to model collision avoidance requirements $\psi = \square\neg$obstacle. Given an LTL specification on the dynamical system (3), it might be impossible for the system to satisfy the specification due to the presence of the adversary, i.e., $\max_\mu \min_\tau Pr^{\mu\tau}_{\mathcal{SG}}(\phi) = 0$. Thus, we relax the specification by allowing violations on invariant constraint ψ. To minimize the impact of invariant constraint violations, we investigate the problem of minimizing the invariant constraint violation rate in this section. In particular, given a specification $\phi = \phi_1 \wedge \psi$, the objective is to compute a control policy that minimizes the expected number of violations of ψ per cycle over all the stationary policies that maximizes the probability of satisfying ϕ_1. We say that every visit to a state that satisfies ϕ_1 completes a cycle. We still focus on the SG generated from (3).

In the following, we first formulate the problem. Motivated by the solution idea of ACPC problem, we solve the problem by generalizing the ACPS problem in [43] and establishing a connection between the problem we formulated and the generalized ACPS problem. Finally, we present the optimality conditions of the problem of interest, and propose an efficient algorithm to solve the problem.

A. Problem Statement

In the following, we focus on how to generate a control policy that minimizes the rate at which ψ is violated while guaranteeing that the probability of satisfying ϕ_1 is maximized. The problem is stated as follows:

Problem 2. *Compute a secure control policy μ that minimizes the violation rate of ψ, i.e., the expected number of violations of ψ per cycle, while maximizing the probability that ϕ_1 is satisfied under any adversary policy τ.*

To investigate the problem above, we assign a positive cost α to every transition initiated from a state s if $s \not\models \psi$. If state $s \models \psi$, we let $g(s) = 0$ for all u_C and u_A. Thus we have for all u_C and u_A

$$g(s) = \begin{cases} \alpha & \text{if } s \not\models \psi \\ 0 & \text{if } s \models \psi. \end{cases} \tag{13}$$

By Proposition 3, we have that two consecutive visits to $\mathcal{E}$ complete a *cycle*. Based on the transition cost defined in (13), Problem 2 can be rewritten as follows.

Problem 3. *Given a stochastic game $\mathcal{SG}$ and an LTL formula ϕ in the form of $\phi = \phi_1 \wedge \psi$, obtain an optimal control policy μ that maximizes the probability of satisfying ϕ_1 while minimizing the average cost per cycle due to violating ψ which is defined as*

$$J^{\mu\tau}_{\mathcal{SG}} = \limsup_{N \to \infty} \mathbb{E}\left\{ \frac{\sum_{k=0}^{N} g(s_k)}{I(\beta, N)} \;\middle|\; \eta_\beta \models \phi_1 \right\}. \tag{14}$$

Since ϕ_1 is required to be satisfied, similar to our analysis in Section V, we first construct a product SG $\mathcal{G}$ using SG $\mathcal{SG}$ and the DRA converted from specification ϕ_1. Then we have the following observations. First, the one-to-one correspondence relationships between the control policies, paths, and associated expected cost due to violating ψ on $\mathcal{SG}$ and $\mathcal{G}$ hold.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2019.2930039, IEEE Transactions on Automatic Control

Furthermore, we observe that if there exists a control policy such that the specification ϕ can be satisfied, it is the optimal solution to Problem 3 with $J^{\mu\tau}_{\mathcal{SG}} = 0$. Finally, by our analysis in Section V, specification ϕ_1 is guaranteed to be satisfied if there exists a control policy that can reach the set of accepting states $\mathcal{E}$. These observations provide us the advantage to analyze Problem 3 on the product SG $\mathcal{G}$ constructed using SG $\mathcal{SG}$ and DRA constructed using ϕ_1. Hence, in the following, we analyze Problem 3 on the product SG $\mathcal{G}$. When the context is clear, we use s to refer to state $(s, q) \in S_{\mathcal{G}}$. Without loss of generality, we assume that $\mathcal{E} = \{1, 2, \cdots, l\}$, i.e., states $\{l+1, \cdots, n\} \cap \mathcal{E} = \emptyset$.

B. Computing the Optimal Control Policy

Due to the presence of an adversary, the results presented in Proposition 1 are not applicable. In the following we first generalize the ACPS problem discussed in [43], which focused on systems without adversaries. Then we characterize the optimality conditions for Problem 3 by connecting it with the generalized ACPS problem.

Generalized ACPS problem. The presence of adversary is not considered in the ACPS problem considered in [43]. Thus we need to formulate the ACPS problem with the presence of adversary and we denote it as the generalized ACPS problem. The objective of generalized ACPS problem is to minimize

$$J_{\mu\tau}(s) = \limsup_{N\to\infty} \frac{1}{N}\mathbb{E}\left\{\sum_{n=0}^{N} g(s) \mid s_0 = s\right\} \tag{15}$$

over all stationary control considering the adverary plays some strategy τ against the controller.

Optimality conditions for generalized ACPS problem. Given any stationary policies μ and τ, denote the induced transition probability matrix as $P^{\mu\tau}$ with $P^{\mu\tau}(s, s') = \sum_{u_C \in U_C(s)} \sum_{u_A \in U_A(s)} \mu(s, u_C)\tau(s, u_A) Pr_{\mathcal{G}}(s, u_C, u_A, s')$. Analogously, denote the expected transition cost starting from any state $s \in S_{\mathcal{G}}$ as $g^{\mu\tau}(s) = \sum_{u_C \in U_C(s)} \sum_{u_A \in U_A(s)} \mu(s, u_C)\tau(s, u_A) g(s)$. Similar to [43], a gain-bias pair is used to characterize the optimality condition. The gain-bias pair $(B^{\mu\tau}, b^{\mu\tau})$ under stationary policies μ and τ, where $B^{\mu\tau}$ is the average cost per stage and $b^{\mu\tau}$ is the differential or relative cost vector, satisfies the following proposition.

Lemma 2. *Let μ and τ be proper stationary policies for a communicating SG, where a communicating SG is an SG whose underlying graph is strongly connected. Then there exists a constant $\zeta^{\mu\tau}$ such that*

$$B^{\mu\tau}(s) = \zeta^{\mu\tau}, \ \forall s \in S_{\mathcal{G}}. \tag{16}$$

Furthermore, the gain-bias pair $(B^{\mu\tau}, b^{\mu\tau})$ satisfies

$$B^{\mu\tau}(s) + b^{\mu\tau}(s) = g^{\mu\tau}(s) + \sum_{k=1}^{n} P^{\mu\tau}(s, k) b^{\mu\tau}(k) \tag{17}$$

Proof. Suppose s' is a recurrent state under policies μ and τ. Define $\xi(s)$ as the expected cost to reach s' for the first time from state s, and $o(s)$ as the expected number of stages to reach s' for the first time from s. Thus $\xi(s')$ and $o(s')$ can be interpretated as the expected cost and expected number of stages to return to s' for the first time from state s', respectively. Based on the definitions above, we have the following equations:

$$\xi(s) = g^{\mu\tau}(s) + \sum_{k \in S \setminus s'} P^{\mu\tau}(s, k)\xi(k), \ \forall s \in S_{\mathcal{G}}, \tag{18}$$

$$o(s) = 1 + \sum_{k \in S_{\mathcal{G}} \setminus s'} P^{\mu\tau}(s, k) o(k), \ \forall s \in S_{\mathcal{G}}. \tag{19}$$

Define $\zeta^{\mu\tau} = \xi(s')/o(s')$. Multiplying (19) by $\zeta^{\mu\tau}$ and subtracting the associated product from (18), we have

$$\begin{aligned}\xi(s) - \zeta^{\mu\tau} o(s) = g^{\mu\tau}(s) - \zeta^{\mu\tau} \\ + \sum_{k \in S_{\mathcal{G}} \setminus s'} P^{\mu\tau}(s, k)(\xi(k) - \zeta^{\mu\tau} o(k)), \ \forall s \in S_{\mathcal{G}}.\end{aligned} \tag{20}$$

Define a bias term

$$b^{\mu\tau}(s) = \xi(s) - \zeta^{\mu\tau} o(s), \ \forall s \in S_{\mathcal{G}} \tag{21}$$

Using (21), (20) can be rewritten as

$$\zeta^{\mu\tau} + b^{\mu\tau}(s) = g^{\mu\tau}(s) + \sum_{k=1}^{n} P^{\mu\tau}(s, k) b^{\mu\tau}(k), \ \forall s \in S_{\mathcal{G}} \tag{22}$$

which completes our proof. □

The result presented above generalizes the one in [43] in the sense that we consider the presence of adversary. The reason that we focus on communicating SG is that we will focus on the accepting states which are strongly connected. Based on Lemma 2, we have the optimality conditions for generalized ACPS problem expressed using the gain-bias pair (B, b):

$$\begin{aligned}B(s) = \min_{\mu} \max_{\tau} \sum_{u_C \in U_C(s)} \sum_{u_A \in U_A(s)} \sum_{s'} \mu(s, u_C)\tau(s, u_A) \\ \cdot Pr_{\mathcal{G}}(s, u_C, u_A, s') B(s')\end{aligned} \tag{23}$$

$$\begin{aligned}B(s) + b(s) = \min_{\mu \in \mu^*} \max_{\tau \in \tau^*} \Big[g^{\mu\tau}(s) + \sum_{u_C \in U_C(s)} \sum_{u_A \in U_A(s)} \sum_{s'} \\ \mu(s, u_C)\tau(s, u_A) Pr_{\mathcal{G}}(s, u_C, u_A, s') b(s')\Big]\end{aligned} \tag{24}$$

where μ^* and τ^* are the optimal policy sets obtained by solving (23). Eq. (23) can be shown using the method presented in Lemma 1, and (24) is obtained directly from (22). Given the optimality conditions (23) and (24) for generalized ACPS problem, we can derive the optimality conditions for Problem 3 by mapping Problem 3 to generalized ACPS problem.

Optimality conditions for Problem 3. In the following, we establish the connection between the generalized ACPS problem and Problem 3. Given the connection, we then derive the optimality conditions for Problem 3.

Denote the gain-bias pair of Problem 3 on the product SG $\mathcal{G}$ as $(J_{\mathcal{G}}, h_{\mathcal{G}})$, where $J_{\mathcal{G}}, h_{\mathcal{G}} \in \mathbb{R}^n$. Denote the gain-bias pair under policies μ and τ as $(J^{\mu\tau}_{\mathcal{G}}, h^{\mu\tau}_{\mathcal{G}})$, where $J^{\mu\tau}_{\mathcal{G}} = [J^{\mu\tau}_{\mathcal{G}}(1), J^{\mu\tau}_{\mathcal{G}}(2), \cdots, J^{\mu\tau}_{\mathcal{G}}(n)]^T$ and $h^{\mu\tau}_{\mathcal{G}} = [h^{\mu\tau}_{\mathcal{G}}(1), h^{\mu\tau}_{\mathcal{G}}(2), \cdots, h^{\mu\tau}_{\mathcal{G}}(n)]^T$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2019.2930039, IEEE Transactions on Automatic Control

We can express the transition probability matrix $P^{\mu\tau}$ induced by control and adversary policy μ and τ as $P^{\mu\tau} = P^{\mu\tau}_{\text{in}} + P^{\mu\tau}_{\text{out}}$, where

$$P^{\mu\tau}_{\text{in}}(s,s') = \begin{cases} P^{\mu\tau}(s,s') & \text{if } s' \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} \tag{25a}$$

$$P^{\mu\tau}_{\text{out}}(s,s') = \begin{cases} P^{\mu\tau}(s,s') & \text{if } s' \notin \mathcal{E} \\ 0 & \text{otherwise} \end{cases}. \tag{25b}$$

Denote the probability that we visit some accepting state $s' \in \mathcal{E}$ from state s under policies μ and τ as $\hat{P}^{\mu\tau}(s,s')$. Then we see that $\hat{P}^{\mu\tau}(s,s')$ is calculated as

$$\begin{aligned}\hat{P}^{\mu\tau}(s,s') = &\sum_{u_C \in U_C(s)} \mu(s,u_C) \sum_{u_A \in U_A(s)} \tau(s,u_A) \\ &Pr_{\mathcal{G}}(s,u_C,u_A,s') + \sum_{u_C \in U_C(s)} \mu(s,u_C) \sum_{u_A \in U_A(s)} \tau(s,u_A) \\ &\sum_{k=l+1}^{n} Pr_{\mathcal{G}}(s,u_C,u_A,k) \hat{P}^{\mu\tau}(k,s'). \end{aligned} \tag{26}$$

The intuition behind (26) is that the probability that s' is the first state to be visited consists of the following two parts. The first term in (26) describes the probability that next state is in $\mathcal{E}$. The second term in (26) models the probability that before reaching state $s' \in \mathcal{E}$, the next visiting state is $k \notin \mathcal{E}$. Denote the transition probability matrix formed by $\hat{P}(s,s')$ as $\hat{P}^{\mu\tau}$. Since $P^{\mu\tau}_{\text{out}}$ is substochastic and transient, we have $I - P^{\mu\tau}_{\text{out}}$ is non-singular [47], where I is the identity matrix with proper dimension. Thus $I - P^{\mu\tau}_{\text{out}}$ is invertible. Then using (25), the transition probability matrix $\hat{P}^{\mu\tau}$ is represented as

$$\hat{P}^{\mu\tau} = (I - P^{\mu\tau}_{\text{out}})^{-1} P^{\mu\tau}_{\text{in}}. \tag{27}$$

Denote the expected invariant property violation cost incurred when visiting some accepting state $s' \in \mathcal{E}$ from state s under policies μ and τ as $\hat{g}(s)$. The expected cost $\hat{g}(s)$ is calculated as follows:

$$\hat{g}(s) = g^{\mu\tau}(s) + \sum_{k=l+1}^{n} Pr^{\mu\tau}(s,k) \hat{g}(k). \tag{28}$$

Under policies μ and τ, denote the expected cost vector formed by $\hat{g}(s)$ as $\hat{g}^{\mu\tau}$. Then using (25), the expected cost vector (28) can be rearranged as follows:

$$\hat{g}^{\mu\tau} = P^{\mu\tau}_{\text{out}} \hat{g}^{\mu\tau} + g^{\mu\tau} \Rightarrow \hat{g}^{\mu\tau} = (I - P^{\mu\tau}_{\text{out}})^{-1} g^{\mu\tau}. \tag{29}$$

Using (27) and (29), we can rewrite (14) as

$$J^{\mu\tau}_{\mathcal{G}} = \limsup_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} \hat{P}^{\mu\tau^k} \hat{g}^{\mu\tau}. \tag{30}$$

Proper policies μ and τ of the product SG $\mathcal{G}$ for Problem 3 are related to proper policies $\hat{\mu}$ and $\hat{\tau}$ for generalized ACPS problem as follows:

$$\hat{P}^{\mu\tau} = P^{\hat{\mu}\hat{\tau}}, \quad \hat{g}^{\mu\tau} = g^{\hat{\mu}\hat{\tau}}, \quad J^{\mu\tau}_{\mathcal{G}} = B^{\hat{\mu}\hat{\tau}}. \tag{31}$$

If we define a bias term $h^{\mu\tau}_{\mathcal{G}} = b^{\hat{\mu}\hat{\tau}}$, then a gain-bias pair $(J^{\mu\tau}_{\mathcal{G}}, h^{\mu\tau}_{\mathcal{G}})$ is constructed for Problem 3. Under the worst case adversary policy $\hat{\tau}$, the control policy that makes the gain-bias pair of ACPS problem satisfy

$$B + b \le g^{\hat{\mu}\hat{\tau}} + P^{\hat{\mu}\hat{\tau}} b \tag{32}$$

is optimal. That is, the control policy μ^* that maps to $\hat{\mu}^*$ is optimal.

To obtain the optimal control policy, we need to characterize Problem 3 in terms of the control and adversary policies μ and τ. The following lemma generalizes the results presented in [9] in which no adversary is considered. For completeness, we show its proof which generalizes the proof in [9].

Lemma 3. *The gain-bias pair $(J^{\mu\tau}_{\mathcal{G}}, h^{\mu\tau}_{\mathcal{G}})$ of Problem 3 under policies μ and τ satisfies the following equations:*

$$J^{\mu\tau}_{\mathcal{G}} = P^{\mu\tau} J^{\mu\tau}_{\mathcal{G}}, \tag{33}$$

$$J^{\mu\tau}_{\mathcal{G}} + h^{\mu\tau}_{\mathcal{G}} = g^{\mu\tau} + P^{\mu\tau} h^{\mu\tau}_{\mathcal{G}} + P^{\mu\tau}_{out} J^{\mu\tau}_{\mathcal{G}}, \tag{34}$$

$$P^{\mu\tau} v^{\mu\tau} = (I - P^{\mu\tau}_{out}) h^{\mu\tau}_{\mathcal{G}} + v^{\mu\tau}, \tag{35}$$

for some vector $v^{\mu\tau}$.

Proof. Given the policies $\hat{\mu}$ and $\hat{\tau}$ for ACPS problem, we have

$$\begin{aligned} J^{\hat{\mu}\hat{\tau}}_{\mathcal{G}} &= P^{\hat{\mu}\hat{\tau}} J^{\hat{\mu}\hat{\tau}}_{\mathcal{G}}, \\ J^{\hat{\mu}\hat{\tau}}_{\mathcal{G}} + h^{\hat{\mu}\hat{\tau}}_{\mathcal{G}} &= g^{\hat{\mu}\hat{\tau}} + P^{\hat{\mu}\hat{\tau}} h^{\hat{\mu}\hat{\tau}}_{\mathcal{G}}, \\ h^{\hat{\mu}\hat{\tau}}_{\mathcal{G}} + v^{\hat{\mu}\hat{\tau}} &= P^{\hat{\mu}\hat{\tau}} v^{\hat{\mu}\hat{\tau}}, \end{aligned}$$

Due to the connection between the control policy of Problem 3 and generalized ACPS problem, we have

$$J^{\mu\tau}_{\mathcal{G}} = P^{\hat{\mu}\hat{\tau}} J^{\mu\tau}_{\mathcal{G}} = (I - P^{\mu\tau}_{\text{out}})^{-1} P^{\mu\tau}_{\text{in}} J^{\mu\tau}_{\mathcal{G}}.$$

By rearranging the equation above, we have $(I - P^{\mu\tau}_{\text{out}}) J^{\mu\tau}_{\mathcal{G}} = J^{\mu\tau}_{\mathcal{G}} - P^{\mu\tau}_{\text{out}} J^{\mu\tau}_{\mathcal{G}} = P^{\mu\tau}_{\text{in}} J^{\mu\tau}_{\mathcal{G}}$. Thus $J^{\mu\tau}_{\mathcal{G}} = (P^{\mu\tau}_{\text{out}} + P^{\mu\tau}_{\text{in}}) J^{\mu\tau}_{\mathcal{G}} = P^{\mu\tau} J^{\mu\tau}_{\mathcal{G}}$. The expression $J^{\mu\tau}_{\mathcal{G}} + h^{\mu\tau}_{\mathcal{G}} = g^{\mu\tau} + P^{\mu\tau} h^{\mu\tau}_{\mathcal{G}} + P^{\mu\tau}_{\text{out}} J^{\mu\tau}_{\mathcal{G}}$ can be rewritten using (27) and (29). We have

$$J^{\mu\tau}_{\mathcal{G}} + h^{\mu\tau}_{\mathcal{G}} = (I - P^{\mu\tau}_{\text{out}})^{-1} (g^{\mu\tau} + P^{\mu\tau}_{\text{in}} h^{\mu\tau}_{\mathcal{G}}).$$

Manipulating the equation above, we see that $(I - P^{\mu\tau}_{\text{out}})(J^{\mu\tau}_{\mathcal{G}} + h^{\mu\tau}_{\mathcal{G}}) = g^{\mu\tau} + P^{\mu\tau}_{\text{in}} h^{\mu\tau}_{\mathcal{G}}$. Then we can see that

$$\begin{aligned} J^{\mu\tau}_{\mathcal{G}} + h^{\mu\tau}_{\mathcal{G}} &= g^{\mu\tau} + (P^{\mu\tau}_{\text{in}} + P^{\mu\tau}_{\text{out}}) h^{\mu\tau}_{\mathcal{G}} + P^{\mu\tau}_{\text{out}} J^{\mu\tau}_{\mathcal{G}} \\ &= g^{\mu\tau} + P^{\mu\tau} h^{\mu\tau}_{\mathcal{G}} + P^{\mu\tau}_{\text{out}} J^{\mu\tau}_{\mathcal{G}}. \end{aligned}$$

Start from $h^{\hat{\mu}\hat{\tau}}_{\mathcal{G}} + v^{\hat{\mu}\hat{\tau}} = P^{\hat{\mu}\hat{\tau}} v^{\hat{\mu}\hat{\tau}}$. We see that $h^{\hat{\mu}\hat{\tau}}_{\mathcal{G}} + v^{\hat{\mu}\hat{\tau}} = (I - P^{\mu\tau}_{\text{out}})^{-1} P^{\mu\tau}_{\text{in}} v^{\mu\tau}$. Therefore we have

$$(I - P^{\mu\tau}_{\text{out}}) h^{\mu\tau}_{\mathcal{G}} + v^{\mu\tau} = P^{\mu\tau} v^{\mu\tau},$$

which completes our proof. □

Lemma 3 indicates that the gain-bias pair can be solved as solutions to a linear system with $3n$ unknowns. Thus we can evaluate any control and adversary policies using Lemma 3, which provides us the potential to implement iterative algorithm to compute the optimal control policy μ.

To compute the control policy μ, we define two operators on $(J_{\mathcal{G}}, h_{\mathcal{G}})$ in (36) and (37), denoted as $T^*(J_{\mathcal{G}}, h_{\mathcal{G}})$ and $T(J_{\mathcal{G}}, h_{\mathcal{G}})$. Generally speaking, we can view them as mappings from $(J_{\mathcal{G}}, h_{\mathcal{G}})$ to $T^*(J_{\mathcal{G}}, h_{\mathcal{G}}) \in \mathbb{R}^n$ and $T_\mu(J_{\mathcal{G}}, h_{\mathcal{G}}) \in \mathbb{R}^n$, respectively. Note that in (37), the transition probability is the one induced under a certain control policy μ.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2019.2930039, IEEE Transactions on Automatic Control

$$(T^*(J_{\mathcal{G}}, h_{\mathcal{G}}))(s) = \min_{\mu} \max_{\tau} \Bigg[\sum_{u_C \in U_C(s)} \sum_{u_A \in U_A(s)} \mu(s,u_C)\tau(s,u_A)g(s) + \sum_{u_C \in U_C(s)} \sum_{u_A \in U_A(s)} \sum_{s'=1}^{n} \mu(s,u_C) \tag{36}$$
$$\cdot \tau(s,u_A) Pr_{\mathcal{G}}(s,u_C,u_A,s') h_{\mathcal{G}}(s') + \sum_{u_C \in U_C(s)} \sum_{u_A \in U_A(s)} \sum_{s'=l+1}^{n} \mu(s,u_C)\tau(s,u_A) Pr_{\mathcal{G}}(s,u_C,u_A,s') J_{\mathcal{G}}(s') \Bigg], \ \forall s$$

$$(T_{\mu}(J_{\mathcal{G}}, h_{\mathcal{G}}))(s) = \max_{\tau} \Bigg[\sum_{u_C \in U_C(s)} \sum_{u_A \in U_A(s)} \mu(s,u_C)\tau(s,u_A)g(s) + \sum_{u_C \in U_C(s)} \sum_{u_A \in U_A(s)} \sum_{s'=1}^{n} \mu(s,u_C)\tau(s,u_A) \tag{37}$$
$$\cdot Pr_{\mathcal{G}}(s,u_C,u_A,s') h_{\mathcal{G}}(s') + \sum_{u_C \in U_C(s)} \sum_{u_A \in U_A(s)} \sum_{s'=l+1}^{n} \mu(s,u_C)\tau(s,u_A) Pr_{\mathcal{G}}(s,u_C,u_A,s') J_{\mathcal{SG}}(s') \Bigg]. \ \forall s$$

Based on the definitions above, we present the optimality conditions for Problem 3 using the following theorem.

Theorem 2. *The control policy μ with gain-bias pair $(J_{\mathcal{G}}^{\mu\tau}, h_{\mathcal{G}}^{\mu\tau})$ that satisfies*

$$J_{\mathcal{G}}^{\mu\tau} + h_{\mathcal{G}}^{\mu\tau} = T^*(J_{\mathcal{G}}^{\mu\tau}, h_{\mathcal{G}}^{\mu\tau}) \tag{38}$$

is the optimal control policy.

Proof. Consider any arbitrary control policy $\hat{\mu}$ and the worst case adversary policy $\hat{\tau}$. By definition of $T^*(\cdot)$ in (36), we have that (38) implies $J_{\mathcal{G}}^{\mu\tau} + h_{\mathcal{G}}^{\mu\tau} \leq g^{\hat{\mu}\hat{\tau}} + P^{\hat{\mu}\hat{\tau}} h_{\mathcal{G}}^{\mu\tau} + P_{\text{out}}^{\hat{\mu}\hat{\tau}} J_{\mathcal{G}}^{\mu\tau}$, where $P^{\hat{\mu}\hat{\tau}}$ and $P_{\text{out}}^{\hat{\mu}\hat{\tau}}$ are the transition probability matrix induced by policies $\hat{\mu}$ and $\hat{\tau}$. Then we have

$$\begin{aligned} J_{\mathcal{G}}^{\mu\tau} + h_{\mathcal{G}}^{\mu\tau} - P_{\text{out}}^{\hat{\mu}\hat{\tau}} J_{\mathcal{G}}^{\mu\tau} &\leq g^{\hat{\mu}\hat{\tau}} + P^{\hat{\mu}\hat{\tau}} h_{\mathcal{G}}^{\mu\tau} \\ &= g^{\hat{\mu}\hat{\tau}} + (P_{\text{in}}^{\hat{\mu}\hat{\tau}} + P_{\text{out}}^{\hat{\mu}\hat{\tau}}) h_{\mathcal{G}}^{\mu\tau}. \end{aligned}$$

Thus we observe that $(I - P_{\text{out}}^{\hat{\mu}\hat{\tau}})(J_{\mathcal{G}}^{\mu\tau} + h_{\mathcal{G}}^{\mu\tau}) \leq g^{\hat{\mu}\hat{\tau}} + P_{\text{in}}^{\hat{\mu}\hat{\tau}} h_{\mathcal{G}}^{\mu\tau}$. Note that $(I - P_{\text{out}}^{\hat{\mu}\hat{\tau}})$ is invertible. Thus the inequality above is rewritten as $J_{\mathcal{G}}^{\mu\tau} + h_{\mathcal{G}}^{\mu\tau} \leq (I - P_{\text{out}}^{\hat{\mu}\hat{\tau}})^{-1}(g^{\hat{\mu}\hat{\tau}} + P_{\text{in}}^{\hat{\mu}\hat{\tau}} h_{\mathcal{G}}^{\mu\tau})$. Rewrite the inequality above according to (27) and (29). Then we have

$$J_{\mathcal{G}}^{\mu\tau} + h_{\mathcal{G}}^{\mu\tau} \leq g^{\tilde{\mu}\tilde{\tau}} + P^{\tilde{\mu}\tilde{\tau}} h_{\mathcal{G}}^{\mu\tau},$$

where $\tilde{\mu}$ and $\tilde{\tau}$ are the control and adversary policies in the associated ACPS problem. Thus, $\tilde{\mu}^*$ satisfies (32) and μ is optimal over all the proper policies. □

Optimal control policy for Problem 3. In the following, we focus on how to obtain an optimal secure control policy. First, note that the optimal control policy consists of two parts. The first part, denoted as μ_{reach}, maximizes the probability of satisfying specification ϕ_1, while the second part, denoted as μ_{cycle}, minimizes the violation cost per cycle due to violating invariant property. Following the procedure described in Algorithm 4, we can obtain the control policy μ_{reach} that maximizes the probability of satisfying specification ϕ_1. Suppose the set of accepting states $\mathcal{E}$ has been reached. Then the control policy μ_{cycle} that optimizes the long term performance of the system is generated using Algorithm 5. Algorithm 5 first initializes the control and adversary policies arbitrarily (e.g., if μ^0 and τ^0 are set as uniform distributions, then $\mu^0(s,u_C) = 1/|U_C(s)|$ and $\tau^0(s,u_A) = 1/|U_A(s)|$ for all s, u_C and u_A). Then it follows a policy iteration procedure to update the control and the corresponding adversary policies until no more improvement can be made. Given μ_{reach} and μ_{cycle}, we can construct the optimal control policy for Problem 3 as

$$\mu^* = \begin{cases} \mu_{\text{reach}}, & \text{if } s \notin \mathcal{E} \\ \mu_{\text{cycle}}, & \text{if } s \in \mathcal{E} \end{cases}. \tag{39}$$

Algorithm 5 Algorithm for a control strategy that minimizes the expected number of invariant constraint violations.

procedure MIN_VIOLATION($\mathcal{G}$, $\mathcal{C}$)
  Input: product SG $\mathcal{G}$, the set GAMECs $\mathcal{C}$ associated with formula ϕ_1
  Output: Control policy μ_{cycle}
  Initialization: Initialize μ^0 and τ^0 be proper policies.
  while $T^*(J_{\mathcal{C}}^{\mu^k\tau^k}, h_{\mathcal{C}}^{\mu^k\tau^k}) \neq T^*(J_{\mathcal{C}}^{\mu^{k-1}\tau^{k-1}}, h_{\mathcal{C}}^{\mu^{k-1}\tau^{k-1}})$ **do**
    Policy Evaluation: Given μ^k and τ^k, calculate the gain-bias pair $(J_{\mathcal{C}}^{\mu^k\tau^k}, h_{\mathcal{C}}^{\mu^k\tau^k})$ using Lemma 3.
    Policy Improvement: Calculate the control policy μ using $\mu = \text{argmin}_{\mu}\ \text{argmax}_{\tau} \left\{ g^{\mu\tau} + P^{\mu\tau} h_{\mathcal{C}}^{\mu^k\tau^k} + P_{\text{out}}^{\mu\tau} J_{\mathcal{C}}^{\mu^k\tau^k} \right\}$.
    Set $\mu^{k+1} = \mu$.
    Set $k = k + 1$.
  end while
end procedure

We finally present the convergence and optimality of Algorithm 5 using the following theorem.

Theorem 3. *Algorithm 5 terminates within a finite number of iterations for any given accepting state set $\mathcal{E}$. Moreover, the result returned by Algorithm 5 satisfies the optimality conditions for Problem 3.*

Proof. In the following, we first prove Algorithm 5 converges within a finite number of iterations. Then we prove that the results returned by Algorithm 5 satisfies the optimality conditions in Theorem 2. We denote the iteration index as k. The control policy at kth iteration is denoted as μ^k. The worst case adversary policy associated with μ^k is denoted as τ^k. Define a vector $\delta \in \mathbb{R}^n$ as $\delta = J_{\mathcal{G}}^{\mu^k\tau^k}\mathbf{1} + h_{\mathcal{G}}^{\mu^k\tau^k} - g^{\mu^{k+1}\tau^{k+1}} - P^{\mu^{k+1}\tau^{k+1}} h_{\mathcal{G}}^{\mu^k\tau^k} - P_{\text{out}}^{\mu^{k+1}\tau^{k+1}} J_{\mathcal{G}}^{\mu^k\tau^k}\mathbf{1}$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2019.2930039, IEEE Transactions on Automatic Control

By Lemma 3, we have $J_{\mathcal{G}}^{\mu^k\tau^k}\mathbf{1}+h_{\mathcal{G}}^{\mu^k\tau^k}=g^{\mu\tau}+P^{\mu\tau}h_{\mathcal{G}}^{\mu^k\tau^k}+P_{\text{out}}^{\mu\tau}J_{\mathcal{G}}^{\mu^k\tau^k}$. By the definition of $T^*(\cdot)$ in (36), the control policy at iteration $k+1$ is computed by optimizing $g^{\mu\tau}+P^{\mu\tau}h_{\mathcal{G}}^{\mu^k\tau^k}+P_{\text{out}}^{\mu\tau}J_{\mathcal{G}}^{\mu^k\tau^k}$. Thus we have that for all s, $\delta(s)\geq 0$. Moreover, we can rewrite vector δ as

$$\begin{aligned}\delta &= J_{\mathcal{G}}^{\mu^k\tau^k}\mathbf{1}+h_{\mathcal{G}}^{\mu^k\tau^k}-g^{\mu^{k+1}\tau^{k+1}}-P^{\mu^{k+1}\tau^{k+1}}h_{\mathcal{G}}^{\mu^{k+1}\tau^{k+1}}\\ &\quad-P_{\text{out}}^{\mu^{k+1}\tau^{k+1}}J_{\mathcal{G}}^{\mu^{k+1}\tau^{k+1}}\mathbf{1}+P^{\mu^{k+1}\tau^{k+1}}h_{\mathcal{G}}^{\mu^{k+1}\tau^{k+1}}\\ &\quad+P_{\text{out}}^{\mu^{k+1}\tau^{k+1}}J_{\mathcal{G}}^{\mu^{k+1}\tau^{k+1}}\mathbf{1}-P^{\mu^{k+1}\tau^{k+1}}h_{\mathcal{G}}^{\mu^k\tau^k}\\ &\quad-P_{\text{out}}^{\mu^{k+1}\tau^{k+1}}J_{\mathcal{G}}^{\mu^k\tau^k}\mathbf{1}\\ &= J_{\mathcal{G}}^{\mu^k\tau^k}\mathbf{1}+h_{\mathcal{G}}^{\mu^k\tau^k}-J_{\mathcal{G}}^{\mu^{k+1}\tau^{k+1}}\mathbf{1}-h_{\mathcal{G}}^{\mu^{k+1}\tau^{k+1}}\\ &\quad-P^{\mu^{k+1}\tau^{k+1}}\left(h_{\mathcal{G}}^{\mu^k\tau^k}-h_{\mathcal{G}}^{\mu^{k+1}\tau^{k+1}}\right)\\ &\quad-P_{\text{out}}^{\mu^{k+1}\tau^{k+1}}\left(J_{\mathcal{G}}^{\mu^k\tau^k}-J_{\mathcal{G}}^{\mu^{k+1}\tau^{k+1}}\right)\mathbf{1},\end{aligned}$$

where the second equality holds by Lemma 3. Thus δ can be represented as

$$\begin{aligned}\delta &= \left(I-P_{\text{out}}^{\mu^{k+1}\tau^{k+1}}\right)\left(J_{\mathcal{G}}^{\mu^k\tau^k}-J_{\mathcal{G}}^{\mu^{k+1}\tau^{k+1}}\right)\mathbf{1}\\ &\quad+\left(I-P^{\mu^{k+1}\tau^{k+1}}\right)\left(h_{\mathcal{G}}^{\mu^k\tau^k}-h_{\mathcal{G}}^{\mu^{k+1}\tau^{k+1}}\right),\end{aligned}\tag{40}$$

where I is the identity matrix. By multiplying $P^{\mu^{k+1}\tau^{k+1}t}$ to both sides of (40) and calculating the summation over t from 0 to $T-1$, we have that

$$\begin{aligned}\sum_{t=0}^{T-1}P^{\mu^{k+1}\tau^{k+1}t}\delta &= \sum_{t=0}^{T-1}P^{\mu^{k+1}\tau^{k+1}t}\left(I-P_{\text{out}}^{\mu^{k+1}\tau^{k+1}}\right)\\ &\cdot\left(J_{\mathcal{G}}^{\mu^k\tau^k}-J_{\mathcal{G}}^{\mu^{k+1}\tau^{k+1}}\right)\mathbf{1}+\sum_{t=0}^{T-1}P^{\mu^{k+1}\tau^{k+1}t}\left(I-P^{\mu^{k+1}\tau^{k+1}}\right)\\ &\quad\cdot\left(h_{\mathcal{G}}^{\mu^k\tau^k}-h_{\mathcal{G}}^{\mu^{k+1}\tau^{k+1}}\right).\end{aligned}\tag{41}$$

Divide both sides by T and let $T\to\infty$. Then we have

$$\begin{aligned}\lim_{T\to\infty}\sum_{t=0}^{T-1}\frac{1}{T}P^{\mu^{k+1}\tau^{k+1}t}\delta &= \lim_{T\to\infty}\sum_{t=0}^{T-1}\frac{1}{T}\big(P^{\mu^{k+1}\tau^{k+1}t}\\ &\quad-P^{\mu^{k+1}\tau^{k+1}t}P_{\text{out}}^{\mu^{k+1}\tau^{k+1}}\big)\big(J_{\mathcal{G}}^{\mu^k\tau^k}-J_{\mathcal{G}}^{\mu^{k+1}\tau^{k+1}}\big)\mathbf{1}\end{aligned}\tag{42}$$

since the second term of (41) is eliminated when $T\to\infty$. Since $P_{\text{out}}^{\mu^{k+1}\tau^{k+1}}$ is a substochastic matrix, we have $P_{\text{out}}^{\mu^{k+1}\tau^{k+1}}\mathbf{1}\leq\mathbf{1}$. Furthermore, since $P^{\mu^{k+1}\tau^{k+1}}$ is a stochastic matrix, we see that $\mathbf{1}-P_{\text{out}}^{\mu^{k+1}\tau^{k+1}}\mathbf{1}\geq 0$. Thus we have $\left(P^{\mu^{k+1}\tau^{k+1}t}-P^{\mu^{k+1}\tau^{k+1}t}P_{\text{out}}^{\mu^{k+1}\tau^{k+1}}\right)\mathbf{1}\geq 0$. Given the inequality above and $\delta\geq 0$, we have that $J_{\mathcal{G}}^{\mu^k\tau^k}-J_{\mathcal{G}}^{\mu^{k+1}\tau^{k+1}}\geq 0$ by observing (42), which implies that $J_{\mathcal{G}}^{\mu^k\tau^k}\geq J_{\mathcal{G}}^{\mu^{k+1}\tau^{k+1}}$.

Consider the scenario where $J_{\mathcal{G}}^{\mu^k\tau^k}=J_{\mathcal{G}}^{\mu^{k+1}\tau^{k+1}}$. We further need to show that in this case $h_{\mathcal{G}}^{\mu^k\tau^k}\leq h_{\mathcal{G}}^{\mu^{k+1}\tau^{k+1}}$. For each state that belongs to the recurrent class, the corresponding entry of $\sum_{t=0}^{T-1}P^{\mu^{k+1}\tau^{k+1}t}$ is positive. By observing (42), we have $\delta(s)=0$ for all s belonging to the recurrent class. Thus according to (41), we have that $h_{\mathcal{G}}^{\mu^k\tau^k}(s)=h_{\mathcal{G}}^{\mu^{k+1}\tau^{k+1}}(s)$ for all s in the recurrent class.

By observing (41), we have

$$\begin{aligned}&\lim_{T\to\infty}\sum_{t=0}^{T-1}P^{\mu^{k+1}\tau^{k+1}t}(h_{\mathcal{G}}^{\mu^k\tau^k}-h_{\mathcal{G}}^{\mu^{k+1}\tau^{k+1}})\\ &= h_{\mathcal{G}}^{\mu^k\tau^k}-h_{\mathcal{G}}^{\mu^{k+1}\tau^{k+1}}-\lim_{T\to\infty}\sum_{t=0}^{T-1}P^{\mu^{k+1}\tau^{k+1}t}\delta\\ &\leq h_{\mathcal{G}}^{\mu^k\tau^k}-h_{\mathcal{G}}^{\mu^{k+1}\tau^{k+1}}-\delta.\end{aligned}$$

Note that the elements corresponding to the transient states in $P^{\mu^{k+1}\tau^{k+1}t}(h_{\mathcal{G}}^{\mu^k\tau^k}-h_{\mathcal{G}}^{\mu^{k+1}\tau^{k+1}})$ approach zero when $t\to\infty$. Thus we have $h_{\mathcal{G}}^{\mu^k\tau^k}(s)-h_{\mathcal{G}}^{\mu^{k+1}\tau^{k+1}}(s)\geq\delta(s)\geq 0$ for all transient states s. Combining all the above together, we have that $\mu^k=\mu^{k+1}$ when $\delta=0$, otherwise $h_{\mathcal{G}}^{\mu^k\tau^k}(s)-h_{\mathcal{G}}^{\mu^{k+1}\tau^{k+1}}(s)\geq 0$ holds for some transient state s.

When Algorithm 5 terminates, we have that

$$T^*(J_{\mathcal{G}}^{\mu^{k+1}\tau^{k+1}},h_{\mathcal{G}}^{\mu^{k+1}\tau^{k+1}})=T^*(J_{\mathcal{G}}^{\mu^k\tau^k},h_{\mathcal{G}}^{\mu^k\tau^k}).\tag{43}$$

By using policy iteration algorithm, the gain-bias pair $(J_{\mathcal{G}}^{\mu^k\tau^k},h_{\mathcal{G}}^{\mu^k\tau^k})$ is first evaluated using Lemma 3 at each iteration k. Then using the gain-bias pair obtained in policy evaluation phase, the T^* operator is calculated as shown in Algorithm 5. Thus according to Lemma 3, we see

$$\mu=\operatorname*{argmin}_{\mu}\max_{\tau}\left\{g^{\mu\tau}+P^{\mu\tau}h_{\mathcal{G}}^{\mu^k\tau^k}+P_{\text{out}}^{\mu\tau}J_{\mathcal{G}}^{\mu^k\tau^k}\right\}.\tag{44}$$

Note that the right hand side of (44) is equivalent to how T^* is calculated in Algorithm 5. Therefore, by combining (43) and (44), we obtain $J_{\mathcal{G}}^{\mu^k\tau^k}+h_{\mathcal{G}}^{\mu^k\tau^k}=T^*(J_{\mathcal{G}}^{\mu^k\tau^k},h_{\mathcal{G}}^{\mu^k\tau^k})$. By Theorem 2, we see that μ^k is the optimal control policy. □

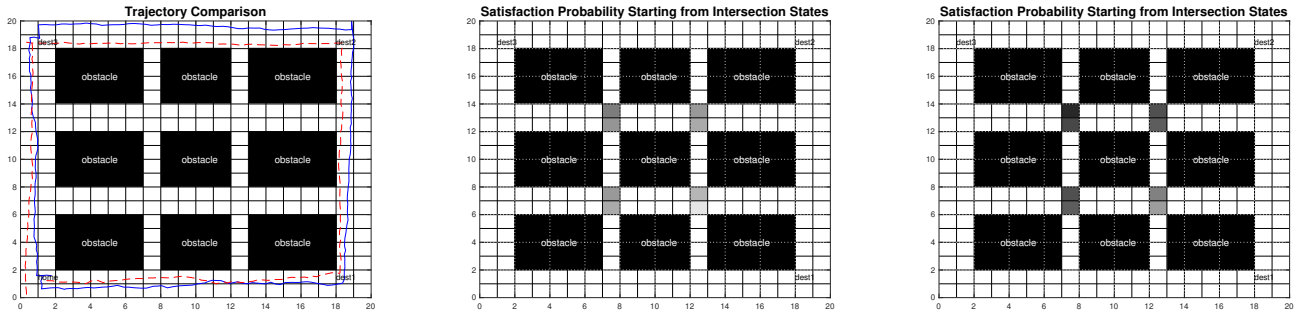
VII. Case Study

In this section, we present two case studies to demonstrate our proposed method. In particular, we focus on the application of remotely controlled UAV under deception attack. In the first case study, the UAV is given a specification modeling reach-avoid requirement. In the second case study, the UAV is given a specification modeling surveillance and collision free requirement. Both case studies were run on a Macbook Pro with 2.6GHz Intel Core i5 CPU and 8GB RAM.

A. Case Study I: Remotely Controlled UAV under Deception Attack with Reach-Avoid Specification

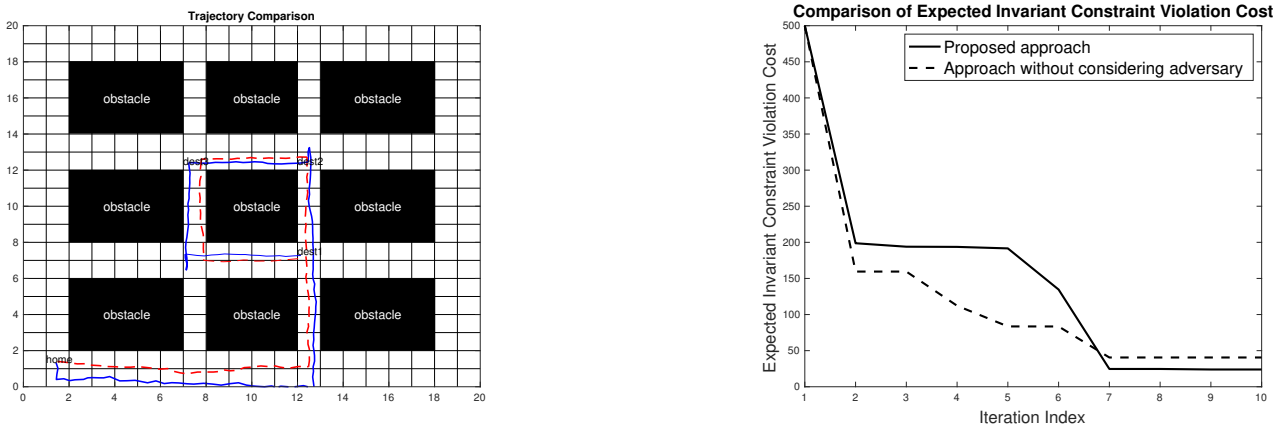
In this case study, we focus on the application of remotely controlled UAV, which conducts package delivery service. The UAV carries multiple packages and is required to deliver the packages to pre-given locations in particular order (e.g., the solution of a travelling salesman problem). The UAV navigates in a discrete bounded grid environment following system model $x(t+1)=x(t)+(u_C(t)+u_A(t)+\vartheta(t))\,\Delta t$, where $x(t)\in\mathcal{R}^2$ is the location of the UAV, $u_C(t)\in\mathcal{U}\subseteq\mathcal{R}^2$ is the control signal, $u_A(t)\in\mathcal{A}\subseteq\mathcal{R}^2$ is the attack signal, $\vartheta(t)\in\mathcal{D}\subseteq\mathcal{R}^2$ is the stochastic disturbance and Δt is time interval. In particular, we let the control set $\mathcal{U}=[-0.3,0.3]^2$, the attack action signal set $\mathcal{A}=[-0.2,0.2]^2$, the disturbance set $\mathcal{D}=[-0.05,0.05]^2$. Also, the disturbance $\vartheta(t)\sim\mathcal{N}(0,\Gamma)$, where $\Gamma=\text{diag}(0.15^2,0.15^2)$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2019.2930039, IEEE Transactions on Automatic Control



(a) (b) (c)

Fig. 1: Comparison of the proposed approach and the approach without considering the presence of the adversary. Fig. 1a gives the trajectories obtained using two approaches. The solid blue line is the trajectory obtained using the proposed approach, while the dashed red line represents the trajectory obtained using the approach without considering the presence of the adversary. Fig. 1b and Fig. 1c present the probability of satisfying the LTL specification using the proposed approach and the approach without considering the adversary when the initial state is set as each of the states lands in the intersections of the grid world, respectively. The shade of gray level at the intersection states corresponds to the satisfaction probability, with black being probability 0 and white being probability 1.



(a) (b)

Fig. 2: Comparison of the proposed approach and the approach without considering the presence of the adversary. Fig. 1a gives the trajectories obtained using two approaches. The solid blue line is the trajectory obtained using the proposed approach, while the dashed red line represents the trajectory obtained using the approach without considering the presence of the adversary. Fig. 2b shows the expected invariant constraint violation cost with respect to iteration indices.

State	$(7,8)$	$(8,8)$	$(13,8)$	$(14,8)$	$(7,13)$	$(8,13)$	$(13,13)$	$(14,13)$
$Pr_{\mathcal{G}}^{\mu_T}$	0.6684	0.6028	0.5915	0.4893	0.8981	0.7126	0.6684	0.6028
$Pr_{\mathcal{G}}^{\mu_{\bar{T}}}$	0.3619	0.3182	0.2878	0.1701	0.6146	0.5112	0.3619	0.3182
Improvement	84.69%	89.44%	105.52%	187.65%	46.13%	39.40%	84.69%	89.44%

TABLE I: Comparison of probabilities of satisfying specification ϕ when starting from the states located in intersections using proposed approach and approach without considering the adversary.

We abstract the system as an SG using Algorithm 1. In particular, given the location of the UAV, we can map the location of the UAV to the grid and simulate the grid it reaches at time $t+1$. Each grid in the environment can be mapped to a state in the SG. In this case study, there exists 400 states in the SG. In the following, we use location and state interchangeably. The control actions and attack signals are sets of discrete control inputs. The label of each state is shown in Fig. 1a. The transition probability can be obtained using Algorithm 1.

The UAV is required to deliver packages to three locations 'dest1', 'dest2', and 'dest3' in this particular order after departing from its 'home'. Then it has to return to 'home' and stay there forever. Also during this delivery service, the UAV should avoid colliding with obstacle areas marked as black in Fig. 1a to Fig. 1c. The LTL formula is written as $\phi = \text{home} \wedge \Diamond(\text{dest1} \wedge \Diamond(\text{dest2} \wedge \Diamond \text{dest3})) \wedge \Diamond\Box \text{home} \wedge \Box\neg \text{obstacle}$.

We compare the control policy obtained using the proposed

approach with that synthesized without considering the presence of the adversary. In Fig. 1a, we present the sample trajectories obtained using these approaches. The solid line shows a sample trajectory obtained by using the proposed approach, and the dashed line shows the trajectory obtained by using the control policy synthesized without considering the presence of adversary. To demonstrate the resilience of the proposed approach, we let the states located in the intersections be labelled as 'home' and hence are set as the initial states. We compare the probability of satisfying the specification ϕ using the proposed approach and the approach without considering the adversary in Fig. 1b and Fig. 1c, respectively. We observe that the control policy synthesized using the proposed approach has higher probability of satisfying specification ϕ. The detailed probability of satisfying specification ϕ is listed in Table I. Denote the probability of satisfying specification ϕ using the proposed approach and the approach without considering the adversary as $Pr_{\mathcal{G}}^{\mu\tau}$ and $Pr_{\mathcal{G}}^{\tilde{\mu}\tilde{\tau}}$, respectively. By using the proposed approach, the average of the improvements of the probability of satisfying the given specification starting from intersection states achieves $(Pr_{\mathcal{G}}^{\mu\tau} - Pr_{\mathcal{G}}^{\tilde{\mu}\tilde{\tau}})/Pr_{\mathcal{G}}^{\tilde{\mu}\tilde{\tau}} = 90.87\%$.

The computation of transition probability took 890 seconds. Given the transition probability, the SG and DRA associated with specification ϕ are created within 1 and 0.01 second, respectively. The computation of product SG took 80 seconds. The product SG has 2000 states and 41700 transitions. It took 45 seconds to compute the control policy.

B. Case Study II: Remotely Controlled UAV under Deception Attack with Liveness and Invariant Specification

In this case study, we focus on the same UAV model as presented in Section VII-A. Let the UAV be given an LTL specification $\phi = \Box(\Diamond(\text{dest1} \wedge \Diamond(\text{dest2} \wedge \Diamond\text{dest3}))) \wedge \Box\neg\text{obstacle}$ consisting of liveness and invariant constraints. In particular, the lieveness constraint $\phi_1 = \Box(\Diamond(\text{dest1} \wedge \Diamond(\text{dest2} \wedge \Diamond\text{dest3})))$ models a surveillance task, i.e., the UAV is required to patrol three critical regions infinitely often following a particular order, and the invariant constraint $\psi = \Box\neg\text{obstacle}$ requires the UAV to avoid collisions with obstacles. Once the critical regions are visited, a cycle is completed. During each cycle, the rate of invariant constraint violation need to be minimized. The cost incurred at each violation is assigned to be 20.

We compare the proposed approach with the approach without considering the adversary. The sample trajectories obtained using these approaches are presented in Fig. 2a. In particular, the solid line shows a sample trajectory obtained by using the proposed approach, and the dashed line shows the trajectory obtained by using the control policy synthesized without considering the presence of adversary. We observe that the control strategy of synthesized using the approach without considering the adversary uses less effort comparing to the proposed approach. However, the proposed approach is more resilient since it uses more control effort to deviate from the obstacles to minimize the violation cost. We present the average invariant constraint violation cost incurred using the control policy obtained at each iteration in Fig. 2b. We observe that the proposed approach incurs lower cost after convergence. In Fig. 2b, the approach that does not consider the adversary incurs lower cost compared to the proposed approach during iterations 2 to 6. The reason is that although the proposed approach guarantees convergence to Stackelberg equilibrium, it does not guarantee optimality of the intermediate policies. The average invariant constraint violation cost using proposed approach is 23.90, while the average invariant constraint violation cost using the approach without considering the adversary is 40.52. The improvement achieved using the proposed approach is 28.67%.

Given the transition probability, the SG and DRA associated with specification ϕ are created within 1 and 0.01 second, respectively. The computation of product SG took 72 seconds. The product SG has 1600 states and 26688 transitions. It took 36 seconds to compute the control policy.

VIII. Conclusion

In this paper, we investigated two problems on a discrete-time dynamical system in the presence of an adversary. We assumed that the adversary can initiate malicious attacks on the system by observing the control policy of the controller and choosing an intelligent strategy. First, we studied the problem of maximizing the probability of satisfying given LTL specifications. A stochastic Stackelberg game was formulated to compute a stationary control policy. A deterministic polynomial-time algorithm was proposed to solve the game. Second, we formulated the problem of minimizing the expected times of invariant constraint violation while maximizing the probability of satisfying the liveness specification. We developed a policy iteration algorithm to compute an optimal control policy by exploiting connections to the ACPS problem. The bottleneck of the proposed framework is the computation complexity of the abstraction procedure. However, this is beyond the scope of this work. The potential ways to reduce the computation complexity include exploring the symmetry of the environment, and applying receding horizon based control framework. In future work, we will consider non-stationary control and adversary policies.

References

[1] M. Lahijanian, S. B. Andersson, and C. Belta, "Temporal logic motion planning and control with probabilistic satisfaction guarantees," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 396–409, 2012.

[2] D. Sadigh and A. Kapoor, "Safe control under uncertainty with probabilistic signal temporal logic," in *Robotics: Science and Systems*, 2016.

[3] T. Wongpiromsarn, U. Topcu, and R. M. Murray, "Receding horizon temporal logic planning for dynamical systems," in *the Proc. of the 48th IEEE Conf. on Decision and Control (CDC)*, 2009, pp. 5997–6004.

[4] S. Karaman and E. Frazzoli, "Sampling-based motion planning with deterministic μ-calculus specifications," in *the Proc. of the 48th IEEE Conf. on Decision and Control (CDC)*, 2009, pp. 2222–2229.

[5] S. G. Loizou and K. J. Kyriakopoulos, "Automatic synthesis of multi-agent motion tasks based on LTL specifications," in *the Proc. of the 43rd IEEE Conf. on Decision and Control (CDC)*, vol. 1, 2004, pp. 153–158.

[6] C. Baier, J.-P. Katoen, and K. G. Larsen, *Principles of Model Checking*. MIT Press, 2008.

[7] M. Kattenbelt and M. Huth, "Verification and refutation of probabilistic specifications via games," in *IARCS Annual Conf. on Foundations of Software Technology and Theoretical Computer Science*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2009.

[8] E. M. Wolff, U. Topcu, and R. M. Murray, "Robust control of uncertain Markov decision processes with temporal logic specifications," in *the Proc. of the 51st IEEE Conf. on Decision and Control (CDC)*, 2012, pp. 3372–3379.

[9] X. Ding, S. L. Smith, C. Belta, and D. Rus, "Optimal control of Markov decision processes with linear temporal logic constraints," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1244–1257, 2014.

[10] K. O'Connell, "CIA Report: Cyber extortionists attacked foreign power grid, disrupting delivery," http://www.ibls.com/internet_law_news_portal_view.aspx?id=1963&s=latestnews.

[11] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage, "Experimental security analysis of a modern automobile," in *IEEE Symp. on Security and Privacy*, 2010, pp. 447–462.

[12] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned aircraft capture and control via GPS spoofing," *Journal of Field Robotics*, vol. 31, no. 4, pp. 617–636, 2014.

[13] P. Paruchuri, J. P. Pearce, J. Marecki, M. Tambe, F. Ordonez, and S. Kraus, "Playing games for security: An efficient exact algorithm for solving Bayesian Stackelberg games," in *Proc. of the Intl. Conf. on Autonomous agents and multiagent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2008, pp. 895–902.

[14] M. Zhu and S. Martinez, "Stackelberg-game analysis of correlated attacks in cyber-physical systems," in *the Proc. of American Control Conf.* IEEE, 2011, pp. 4063–4068.

[15] N. Basilico, N. Gatti, and F. Amigoni, "Patrolling security games: Definition and algorithms for solving large instances with single patroller and single intruder," *Artificial Intelligence*, vol. 184, pp. 78–123, 2012.

[16] M. Tambe, *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press, 2011.

[17] T. Chen, V. Forejt, M. Z. Kwiatkowska, D. Parker, and A. Simaitis, "Prism-games: A model checker for stochastic multi-player games." in *the Proc. of Intl. Conf. on TACAS*. Springer, 2013, pp. 185–191.

[18] J. Ding, M. Kamgarpour, S. Summers, A. Abate, J. Lygeros, and C. Tomlin, "A stochastic games framework for verification and control of discrete time stochastic hybrid systems," *Automatica*, vol. 49, no. 9, pp. 2665–2674, 2013.

[19] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas, "Where's Waldo? sensor-based temporal logic motion planning," in *the Proc. of IEEE Intl. Conf. on Robotics and Automation*, 2007, pp. 3116–3121.

[20] A. Bhatia, L. E. Kavraki, and M. Y. Vardi, "Sampling-based motion planning with temporal goals," in *the Proc. of IEEE Intl. Conf. on Robotics and Automation*, 2010, pp. 2689–2696.

[21] E. Plaku, L. E. Kavraki, and M. Y. Vardi, "Motion planning with dynamics by a synergistic combination of layers of planning," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 469–482, 2010.

[22] J. Fu, N. Atanasov, U. Topcu, and G. J. Pappas, "Optimal temporal logic planning in probabilistic semantic maps," in *the Proc. of IEEE Intl. Conf. on Robotics and Automation*, 2016, pp. 3690–3697.

[23] M. Lahijanian, J. Wasniewski, S. B. Andersson, and C. Belta, "Motion planning and control from temporal logic specifications with probabilistic satisfaction guarantees," in *the Proc. of IEEE Intl. Conf. on Robotics and Automation*, 2010, pp. 3227–3232.

[24] M. Kloetzer and C. Belta, "A fully automated framework for control of linear systems from temporal logic specifications," *IEEE Transactions on Automatic Control*, vol. 53, no. 1, pp. 287–297, 2008.

[25] J. Fu and U. Topcu, "Synthesis of shared autonomy policies with temporal logic specifications," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 1, pp. 7–17, 2016.

[26] V. Raman and H. Kress-Gazit, "Analyzing unsynthesizable specifications for high-level robot behavior using LTLMoP," in *Computer Aided Verification*. Springer, 2011, pp. 663–668.

[27] J. Tumová, L. I. R. Castro, S. Karaman, E. Frazzoli, and D. Rus, "Minimum-violation LTL planning with conflicting specifications," in *the Proc. of American Control Conf.* IEEE, 2013, pp. 200–205.

[28] A. Nilim and L. El Ghaoui, "Robust control of Markov decision processes with uncertain transition matrices," *Operations Research*, vol. 53, no. 5, pp. 780–798, 2005.

[29] J. Fu and U. Topcu, "Synthesis of joint control and active sensing strategies under temporal logic constraints," *IEEE Transactions on Automatic Control*, vol. 61, no. 11, pp. 3464–3476, 2016.

[30] D. Fudenberg and J. Tirole, *Game Theory*. MIT Press, 1991.

[31] T. Quatmann, C. Dehnert, N. Jansen, S. Junges, and J.-P. Katoen, "Parameter synthesis for Markov models: Faster than ever," in *Intl. Symp. on Automated Technology for Verification and Analysis*. Springer, 2016, pp. 50–67.

[32] M. Kattenbelt, M. Kwiatkowska, G. Norman, and D. Parker, "A game-based abstraction-refinement framework for Markov decision processes," *Formal Methods in System Design*, vol. 36, no. 3, pp. 246–280, 2010.

[33] L. de Alfaro and T. A. Henzinger, "Concurrent omega-regular games," in *Proc. of IEEE symp. on Logic in Computer Science*, 2000, pp. 141–154.

[34] C. Y. Ma, N. S. Rao, and D. K. Yau, "A game theoretic study of attack and defense in cyber-physical systems," in *the Proc. of IEEE Conf. on Computer Communications Workshops*, 2011, pp. 708–713.

[35] Y. Li, L. Shi, P. Cheng, J. Chen, and D. E. Quevedo, "Jamming attacks on remote state estimation in cyber-physical systems: A game-theoretic approach," *IEEE Transactions on Automatic Control*, vol. 60, no. 10, pp. 2831–2836, 2015.

[36] D. Korzhyk, Z. Yin, C. Kiekintveld, V. Conitzer, and M. Tambe, "Stackelberg vs. Nash in security games: An extended investigation of interchangeability, equivalence, and uniqueness," *Journal of Artificial Intelligence Research*, vol. 41, no. 2, pp. 297–327, 2011.

[37] L. Niu and A. Clark, "Secure control under linear temporal logic constraints." in *the Proc. of American Control Conf.* IEEE, 2018.

[38] Y. Shoukry and P. Tabuada, "Event-triggered state observers for sparse sensor noise/attacks," *IEEE Transactions on Automatic Control*, vol. 61, no. 8, pp. 2079–2091, 2016.

[39] H. Fawzi, P. Tabuada, and S. Diggavi, "Secure estimation and control for cyber-physical systems under adversarial attacks," *IEEE Transactions on Automatic Control*, vol. 59, no. 6, pp. 1454–1467, 2014.

[40] M. H. Manshaei, Q. Zhu, T. Alpcan, T. Bacşar, and J.-P. Hubaux, "Game theory meets network security and privacy," *ACM Computing Surveys*, vol. 45, no. 3, p. 25, 2013.

[41] Q. Zhu and T. Basar, "Game-theoretic methods for robustness, security, and resilience of cyberphysical control systems: games-in-games principle for optimal cross-layer resilient control systems," *IEEE Control Systems Magazine*, vol. 35, no. 1, pp. 46–65, 2015.

[42] Q. Zhu and T. Başar, "Robust and resilient control design for cyber-physical systems with an application to power systems," in *the Proc. of the 50th IEEE Conf. on Decision and Control and European Control Conf.*, 2011, pp. 4066–4071.

[43] D. P. Bertsekas, D. P. Bertsekas, D. P. Bertsekas, and D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific Belmont, MA, 1995, vol. 1, no. 2.

[44] M. Lahijanian, S. B. Andersson, and C. Belta, "A probabilistic approach for control of a stochastic system from LTL specifications," in *the Proc. of the 48th IEEE Conf. on Decision and Control/Chinese Control Conf.*, 2009, pp. 2236–2241.

[45] O. Cappé, S. J. Godsill, and E. Moulines, "An overview of existing methods and recent advances in sequential monte carlo," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 899–924, 2007.

[46] J. v. Neumann, "Zur theorie der gesellschaftsspiele," *Mathematische annalen*, vol. 100, no. 1, pp. 295–320, 1928.

[47] L. Hogben, *Handbook of Linear Algebra*. Chapman and Hall/CRC, 2013.

Luyao Niu (SM'15) received the B.Eng. degree from the School of Electro-Mechanical Engineering, Xidian University, Xian, China, in 2013 and the M.Sc. degree from the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute (WPI) in 2015. He has been working towards his Ph.D. degree in the Department of Electrical and Computer Engineering at Worcester Polytechnic Institute since 2016. His current research interests include optimization, game theory, and control and security of cyber physical systems.

Andrew Clark (M'15) is an Assistant Professor in the Department of Electrical and Computer Engineering at Worcester Polytechnic Institute. He received the B.S. degree in Electrical Engineering and the M.S. degree in Mathematics from the University of Michigan - Ann Arbor in 2007 and 2008, respectively. He received the Ph.D. degree from the Network Security Lab (NSL), Department of Electrical Engineering, at the University of Washington - Seattle in 2014. He is author or co-author of the IEEE/IFIP William C. Carter award-winning paper (2010), the WiOpt Best Paper (2012), and the WiOpt Student Best Paper (2014), and was a finalist for the IEEE CDC 2012 Best Student-Paper Award. He received the University of Washington Center for Information Assurance and Cybersecurity (CIAC) Distinguished Research Award (2012) and Distinguished Dissertation Award (2014). His research interests include control and security of complex networks, submodular optimization, and control-theoretic modeling of network security threats.